Transfert sécurisé d'Images par combinaison de techniques de compression, cryptage et de marquage

José Marconi Rodrigues

ACADÉMIE DE MONTPELLIER
UNIVERSITÉ MONTPELLIER II
- SCIENCES ET TECHNIQUES DU LANGUEDOC -

THÈSE

présentée pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ MONTPELLIER II
Mention INFORMATIQUE

par

José Marconi M. RODRIGUES

Discipline : Informatique - Image
Formation Doctorale : Informatique
École Doctorale : Information, Structures et Systèmes

Titre de la thèse :

TRANSFERT SÉCURISÉ D'IMAGES PAR COMBINAISON DE TECHNIQUES DE COMPRESSION, CRYPTAGE ET MARQUAGE

Soutenue publiquement le 31 Octobre 2006

JURY

M. Jean-Claude BAJARD	Professeur, Univ.Montpellier II, LIRMM	Directeur de thèse
M. Adrian BORS	Associate Professor, University of York, UK	Examinateur
M. Christian ROUX	Professeur, GET - ENST Bretagne	Rapporteur
M. Christophe FIORIO	MC, Univ.Montpellier II, LIRMM	Co-directeur
M. Jean-Claude KONIG	Professeur, Univ.Montpellier II, LIRMM	Examinateur
M. Jean-Eric DEVELAY-MORIC	Médecin, CHU Montpellier-Nîmes	Examinateur
M. Jean-Pierre GUEDON	Professeur, IRCCyN Polytech Nantes	Rapporteur
M. William PUECH	MC HDR, CUFRN, LIRMM	Co-directeur

A mes parents bien-aimés, Sobrinho et Lucy, qui n'ont jamais baissé les bras devant les difficultés de la vie et à mes filles Karine et Rebecca qui ont compris et accepté mon absence durant ces trois ans. Cette thèse est pour vous.

Remerciements

La première personne que je souhaite remercier est mon codirecteur de thèse William Puech. Il m'a constamment soutenu, encouragé et stimulé pendant ces trois années de thèse. Ses nombreuses remarques ont montré un très vaste connaissance des sujets abordés et m'ont donné les conduits et les améliorations de mon travail. En dehors du plan technique, nous avons eu un rapport d'une très forte amitié et de respect. Je remercie également le professeur Jean-Claude Bajard et Christophe Fiorio mon directeur et codirecteur de thèse respectivement pour ses remarques aussi pertinentes.

Ensuite je voudrais remercier les membres de mon jury d'avoir accepté d'évaluer mon travail. En particulier je remercie les Messieur Christian Roux et Jean-Pierre Guedon d'avoir accepté de rapporter ma thèse, et de l'intérêt qu'ils y ont porté. J'ai conscience de la lourde charge de travail que cela a représenté.

Je souhaite remercie les membres d'équipe ICAR particulièrement (Marc Chaumont, Jean Triboulet et Gilles Gesquier) pour leurs commentaires toujours encourageant pour mon travail. Olivier Strauss pour son regard critique pertinent à tout travail scientifique.

La suite des remerciements est destinée à mon entourage. Mickaél Sauvee et Gael Pages, qui m'ont permis de libérer la pression au quotidien. Fabien Lydoire, Michel Dominici et Aurelien Noce en particulier pour des réponses à mes problèmes avec la langue française. Jean-Mathias qui a apporté sa bonne humeur dans leur travail. Merci pour la super soirée après le pot de thèse. Philippe Amat toujours prêt à aider n'importe pas dans quelle situation. Samer Mouhammed pour venir dans mon bureaux dire bonjour et m'encourager pour la thèse et présentation. Vicent Nabat, Wallid Zarrad et David Corbel pour leur très agréable companie au répas de midi. Carla Aguiar, la brésilienne, pour son esprit chaleureux.

Un grand merci à ceux qui assurent le bon fonctionnement du laboratoire. Nicole Olivet, Céline Berger, Louise Casteill, Cécile Lukasik, Elisabeth Petiot, Patrice Prouha et Nadine Tilloy pour leurs travaux et pour leurs dévouements.

En dehors de mon environnement de travail, j'ai eu le plaisir de faire la connaissance de personnes formidable. Une famille merveilleuse : Carla et Loriane (les enfants) vous allez beaucoup me manquer. Magali (la maman) toujours avec un grand sourire, merci d'avoir lu ma thèse et corriges mes fautes de français. Je remercie Vincent Baily et Veronique de m'avoir fait découvrir des petites villages en France. Je remercie mes amis Jean-Claude, Maggy, M.Odet et Mme Odette pour ses sincères amitiés et pour me soutenir dans les moments difficiles.

Je ne peux terminer ces remerciements sans mentionner mes proches Francisco Ro-

drigues, Maria Lucy, Karine Rodrigues et Rebecca Rodrigues ainsi que toute ma famille, à qui je dédie ce travail. Merci par votre confiance.

Résumé

Les réseaux numériques ont fortement évolué ces dernières années et sont devenus inévitables pour la communication moderne. Les images transmises sur ces réseaux sont des données particulières du fait de leur quantité importante d'information. La transmission des images soulève donc un nombre important de problèmes qui ne sont pas encore tous résolus. Nous citons, par exemple, la sécurité, la confidentialité, l'intégrité et l'authenticité des images pendant leur transmission. Certaines applications médicales exigent une association stricte entre l'image et des données contextuelles. La protection des informations haute résolution (hautes fréquences des images, détails, visualisation réglable) connaît actuellement une demande forte. Durant cette thèse, nos travaux de recherche ont conduit à la création de trois nouvelles approches permettant de sécuriser le transfert d'images. Les deux premières méthodes s'appuient sur des codages hybrides : emploi conjoint de cryptage, insertion de données cachées et compression. La troisième approche s'appuie sur le travail de Droogenbroeck et Benedett. Nous proposons un cryptage sélectif pour protéger la transmission d'images. Il permet de crypter de manière sélective l'image en protégeant les informations des détails tout en restant compatible aux standards de compression d'images.

Abstract

The networks have been developing strongly these last years and became inevitable for the modern communication. The images transmitted on these networks are particular data because of their large quantity of information. The transmission of the images thus raises a significant number of problems which the majority of them have no solution yet. We enumerate for example safety, confidentiality, integrity and the authenticity of these data during their transmission. Some medical applications require a strict association between the image and its contextual data. The protection of high resolution information (high frequencies of the images, details, scalable visualization) currently has a high demand. During this thesis, our research led to the creation of three new methods to make safe the transfer of images. The first two methods are based on hybrid codings : use of encryption, data hidden and compression. The third approach is based on the work of Droogenbroeck and Benedett. We propose selective encryption to protect the transmission of images. It allows to selectively encrypt the image by protecting information from high frequencies while remaining compatible according to the standards of image compression.

Table des matières

Remerciements **1**

Table des matières **2**

Introduction Générale **7**

I État de l'art **11**

1 Cryptage **15**

Introduction 16

1.1 Généralités sur le cryptage 16

1.1.1 Cryptosystèmes classiques 16

1.1.2 Classification des algorithmes 17

1.1.3 Notion de sécurité 18

1.2 Algorithmes de chiffrement et clefs 19

1.2.1 Chiffrement asymétrique 19

1.2.2 Chiffrement symétrique 22

1.2.3 Chiffrement hybride 23

1.2.4 Quelques remarques sur les tailles des clefs 24

1.3 Chiffrement par flot 25

1.3.1 Chiffrement de Vernam (One-time pad) 25

1.3.2 Chiffrement synchrone 25

1.3.3 Chiffrement asynchrone 26

1.4 Chiffrement par bloc 27

1.4.1 Algorithme DES et 3-DES 28

1.4.2 Standard AES 28

1.4.3 Modes d'opération 29

1.5 Autres cryptosystèmes 31

1.5.1 Approches mathématiques 31

1.5.2 Approches physiques 32

1.6 Cryptanalyse 32

Conclusion 33

2 Insertion de données cachées **35**
Introduction . . . 36
2.1 Généralités sur l'IDC . . . 37
2.1.1 Conditions requises . . . 37
2.1.2 Domaines d'insertion . . . 38
2.1.3 Mesures et modèles perceptueles . . . 39
2.2 Stéganographie . . . 41
2.2.1 Classifications des techniques de stéganographie . . . 41
2.3 Marquage d'image fixe . . . 43
2.3.1 Classification des techniques de marquage . . . 43
2.3.2 Modèles perceptifs pour le marquage d'images fixes . . . 45
2.3.3 Manipulations et attaques sur les images . . . 46
2.3.4 Techniques robustes aux distorsions synchrones . . . 48
2.3.5 Techniques robustes aux distorsions asynchrones . . . 49
Conclusion . . . 51

3 Compression d'image **53**
Introduction . . . 54
3.1 Généralités sur la compression et l'images . . . 54
3.1.1 Taux de compression et redondance . . . 54
3.1.2 Critères psychovisuels et compression . . . 55
3.2 Codage sans perte . . . 56
3.2.1 Codage d'Huffman . . . 56
3.2.2 Codage arithmétique . . . 56
3.2.3 L'algorithme LZW . . . 57
3.2.4 Codage par plage . . . 58
3.2.5 Codage par prédiction linéaire . . . 58
3.3 Codage avec pertes . . . 59
3.3.1 Quantification . . . 59
3.3.2 Codage prédictif avec pertes . . . 59
3.3.3 Codage par transformation . . . 60
3.4 Domaine spatial . . . 60
3.4.1 Quadtree . . . 60
3.4.2 Décomposition en plans binaire . . . 61
3.4.3 Fractales . . . 61
3.5 Domaine fréquentiel . . . 62
3.5.1 Les standards . . . 62
Conclusion . . . 66

II Transfert sécurisé d'images par combinaison de techniques de cryptage, marquage et compression **69**

4 Codage Hybride **73**

4.1 Méthode réversible de crypto-tatouage appliquée aux images médicales . 74
Introduction . . . 74
4.1.1 Décomposition d'image . . . 75
4.1.2 Compression sans perte . . . 76
4.1.3 Insertion de données cachées . . . 79
4.1.4 Brouillage et cryptage sélectif . . . 80
4.1.5 Décodage . . . 81
4.1.6 Résultats . . . 83
4.1.7 Bilan . . . 85
4.2 Une méthode autonome de masquage de données . . . 87
Introduction . . . 87
4.2.1 Nouvelle méthode combinant cryptage et IDC . . . 89
4.2.2 Algorithme de chiffrement par flot asynchrone proposé . . . 90
4.2.3 L'insertion de données cachées dans le domaine spatial . . . 95
4.2.4 Résultats . . . 95
4.2.5 Cryptanalyse . . . 97
4.2.6 Bilan . . . 100
Conclusion . . . 101

5 Cryptage Sélectif **103**
Introduction . . . 104
5.1 Codage entropique du JPEG . . . 104
5.2 Chiffrement AES en mode CFB . . . 108
5.3 Travaux précédents de CS par DCT . . . 109
5.4 L'approche proposée . . . 110
5.4.1 Quelques considérations . . . 110
5.4.2 Cryptage sélectif . . . 113
5.4.3 Procédure de décryptage . . . 115
5.5 Résultats . . . 116
5.5.1 Application à l'imagerie médicale . . . 116
5.5.2 Application de notre méthode à une BDD de peintures numériques 119
5.5.3 Application de notre méthode à la protection des visages dans des séquences d'images . . . 122
5.6 Bilan . . . 122
Conclusion . . . 124

Conclusion Générale **125**

A Transformée cosinus discrète **129**

B Entropie **131**

Bibliographie **146**

Table des figures **147**

Introduction Générale

Définition du problème

Les réseaux numériques ont tellement évolué qu'ils sont devenus un mécanisme essentiel de communication. Ils permettent de transmettre toute sorte d'informations textuelles, sonores et principalement des images. La croissance exponentielle du trafic des images est renforcée par l'apparition importante d'appareils photos numérique et l'utilisation des téléphones portables.

Les images sont des données particulières du fait de la quantité importante d'information et de leur disposition bidimensionnelle. La transmission des images soulève donc un nombre conséquent de problèmes qui ne sont pas tous encore résolus. De plus, les réseaux informatiques sont complexes et les écoutes illégales nombreuses.

Afin de comprendre l'approche de notre travail de recherche, nous avons classé les problématiques abordées en deux groupes.

Premier groupe de problématique

Dans ce premier groupe nous avons réuni les problèmes concernant la sécurité et l'authenticité des images transmises sur des réseaux numériques.

- Un premier problème est relatif à l'aspect sécurité et à l'authenticité des données pendant la transmission, mais également après réception de celles-ci. Toute information circulant peut être capturée, lue et/ou modifiée.
- Un deuxième problème concerne le temps de transfert. En effet, du fait de la quantité importante de données, les tailles des images doivent être réglable avant le transfert.
- Des applications exigent que l'image soit associée à des informations contextuelles. Il ne faut absolument pas que ces informations puissent être dissociées. C'est le cas des images médicales et les diagnostics des patients.
- Pour des raisons de confidentialité, certaines images doivent être rendues complètement ou partiellement illisibles et non déchiffrables pendant le transfert.

Second groupe de problématique

Une autre problématique que nous avons également abordée et la visualisation et la protection réglable des données contenues dans les images.

- Certaines applications demandent que l'image soit visible, mais que les données qui définissent les détails de l'image soient protégées. C'est le cas des œuvres d'art numériques qui doivent préserver l'image originale.
- Les images médicales prises avec des appareils mobiles sur les lieux des accidents doivent être transmises rapidement et de façon sûre en préservant la confidentialité du patient.
- La protection de la vie privée et des libertés exige que certaines régions dans l'image ne soient pas identifiables. C'est le cas de la dissimulation des visages pour la vidéo-surveillance.

Contribution

Dans ces travaux nous allons présenter trois méthodes que nous avons développées au cours de cette thèse. Les deux premières méthodes concernent la problématique décrite dans le premier groupe. La dernière méthode a été conçue pour affronter les problèmes exposés dans le second groupe.

Contribution à la problématique du premier groupe

Plusieurs méthodes ont été développées pour résoudre les problèmes de sécurité et d'authenticité dans le transfert des images. Toutefois, très peus méthodes proposent à la fois une compression, un chiffrement et une IDC (insertion de données cachées). Parmi eux, nous citons le travail de Autrusseau [Aut02] qui exploite des critères psychovisuels et les propriétés de la transformation Mojette pour le tatouage et a compression d'image. Nous citons également le travail de Borie [Bor04] qui applique de codage hybride pour la sécurisation d'images médicales. Il nous a donc semblé nécessaire de trouver des nouvelles méthodes permettant un codage hybride. C'est-à-dire l'emploi conjoint du cryptage, de l'IDC et de la compression. L'utilisation de codage hybride pour la sécurisation du transfert d'images est encore aujourd'hui très innovant.

Avec cette approche de codage hybride, nous avons développé une méthode qui permet d'insérer une grande quantité d'information dans l'image sans augmenter la taille de celle-ci et sans en modifier le contenu tout en la chiffrant.

Nous avons aussi utilisé l'approche de codage hybride pour développer également une autre méthode pour le transfert autonome et sécurisé d'images. Nous proposons l'utilisation de plusieurs catégories de cryptage et l'IDC dans l'image cryptée afin de rendre autonome un système de transmission sécurisé d'images.

Contribution à la problématique du second groupe

Nous proposons une nouvelle méthode permettant de crypter de manière sélective et partielle les données de l'image tout en conservant un niveau de sécurité et de visibilité

suffisant. Notre approche de cryptage sélectif des données est effectué en même temps que la compression et ne rajoute aucune donnée supplémentaire.

Plan du manuscrit

Ce manuscrit est composé de deux parties et est organisé de la façon suivante. La première partie établit un état de l'art des différents domaines dans lesquels s'inscrivent ces travaux de thèse. Cette première partie se décompose en trois chapitres. Le premier chapitre présente les classifications des différentes techniques de cryptage. Ensuite, le chapitre 2 décrit les aspects principaux et les technologies pour la dissimulation d'information, plus précisément l'insertion de données cachées (IDC), la stéganographie et le marquage d'images. Le chapitre 3 expose les concepts et le développement des techniques de compression d'images.

La seconde partie est composée de deux chapitres et présente les trois méthodes développées durant la thèse. Le chapitre 4 détaille les deux premières méthodes de codage hybride qui ont été développées pour faire face aux problématiques décrites dans le premier groupe. Le chapitre 5 est consacré au cryptage sélectif et partiel concernant les problématiques du second groupe. Dans chaque chapitre de la seconde partie, nous présentons des exemples et résultats expérimentaux sur des images réelles. Nous avons aussi effectué des bilans en faisant ressortir les avantages et faiblesses des méthodes développées.

Première partie

État de l'art

Première partie

État de l'art

L'objectif de cette partie est de présenter les aspects principaux et les méthodes liés aux évolutions des technologies traitées dans le contexte de la thèse. Cette partie est composée de trois chapitres, les deux premiers abordent la dissimulation/protection d'information et le dernier la compression d'images. Le premier chapitre présente les aspects de la cryptographie, quelques théories, les algorithmes de chiffrement et des nouvelles techniques de cryptage. Le deuxième chapitre expose les concepts et techniques d'insertion de données cachées (IDC) dans la stéganographie et dans le marquage d'image. Le troisième chapitre, enfin, expose le développement des techniques de compression d'images.

Introduction première partie

L'information est un élément constitutif et déterminant dans tous les domaines. Tout au long de l'histoire, l'humanité a essayé d'envoyer des informations d'une façon sécurisée. La dissimulation d'information a été utilisée comme instrument de sécurisation pour les stratégies militaires et échange de données secrètes [KP00, Kah96, Kob90]. La nécessité de transfert sécurisé d'information a traversé le temps et est encore énormément utilisée dans le monde numérique. Il y a une variété importante de domaines d'applications des méthodes de dissimulation d'information. Ces méthodes disposent de différentes caractéristiques et peuvent être classifiées selon leurs objectifs et leurs contraintes. D'une façon générale, les méthodes de dissimulation d'information peuvent être groupées en deux grandes familles. La première famille utilise des techniques pour rendre incompréhensible le message : la **cryptographie** qui se sert des théories mathématiques pour rendre le message indéchiffrable, et le **brouillage** où mélange (*scrambling*) qui utilise des techniques pour désordonner le message et le rendre illisible. La deuxième famille utilise une porteuse comme « enveloppe » pour cacher le message. Nous distinguons la **stéganographie** qui cherche la communication invisible. Le **marquage** (*watermarking*) qui concerne l'insertion d'une marque (filigrane) qui doit être robuste et qui a comme but la protection de la propriété intellectuelle. L'IDC **insertion de données cachées** (*data hiding*) qui est un marquage avec une quantité importante de données. L'**empreinte digitale** qui est une forme de marquage où chaque objet reçoit un numéro d'identification connu et unique. La **signature numérique** est aussi une marque qui dépend simultanément des informations obtenues du document clair, à partir des *fonctions d'hachages*[1], et des informations d'auteurs.

Tout au long de ce document, nous faisons référence à des notions pertinentes des méthodologies de dissimulation d'information et de compression d'images. Nous énumérons donc certains termes et terminologies qui faciliteront la compréhension des concepts et des objectifs des travaux de recherche développés dans les parties suivantes.

Les transmissions croissantes des informations dans les réseaux publics ont apporté divers **problèmes relatifs à la sécurité des données**.

Confidentialité

Problème : Toute information circulant peut être capturée et lue (*sniffing*).
La confidentialité se base sur les concepts qui permettent de s'assurer que l'information ne puisse pas être lue par des personnes non autorisées. La confidentialité est fortement liée à la cryptographie, présentée chapitre 1.

Authentification

Problème : Une personne peut falsifier ses informations numériques personnelles (*spoofing*).
L'authentification est l'ensemble de moyens qui permet d'assurer que les données reçues

[1] Fonction irréversible appliquée à un document pour produire une suite compactée de bits qui le caractérise.

et envoyées proviennent bien des entités déclarées. Nous citons la reconnaissance biométrique, le certificat numérique et l'empreinte digitale comme les instruments les plus utilisés pour l'authentification des personnes.

Intégrité des données
Problème : Les données peuvent être capturées et modifiées.
L'intégrité des données concerne les techniques qui rendent possible la vérification de la non-altération des données, c'est-à-dire le contrôle du contenu. Le marquage fragile, présenté chapitre 2, est un des instruments de contrôle d'intégrité des données.

Non-répudiation
Problème : Dans certains échanges électroniques, il n'existe pas de témoignage de participation.
La non-répudiation est la façon d'empêcher à une entité (émetteur ou récepteur) de nier la participation dans un échange de données. Les systèmes d'échange avec clé publique, chapitre 1, et signature numérique assurent la non-répudiation.

Dans la dissimulation d'information les **attaques** ont un rôle important et il existe différentes classifications selon leur objectifs et moyens. Des attaques dans la cryptographie (cryptanalyse), section 1.6, sont les manières d'essayer d'obtenir la clef à partir d'un texte chiffré ou de déchiffrer un texte sans avoir la clef. Des attaques dans l'IDC sont les façons de détecter, supprimer, modifier ou extraire le message ou la marque, présenté chapitre 2.

La compression d'images numériques, présenté chapitre 3, a pour but de réduire le nombre de bits qui représent l'image. D'une façon générale, la compression peut être schématisée en trois phases : pré-traitement, quantification et codage.

Pré-traitement
Au niveau du pré-traitement, il s'agit d'appliquer des transformations (espaces couleur et/ou espaces fréquentiels) pour décorreler les redondances spatiales entre les éléments voisins de l'image.

Quantification
La quantification a pour objectif de réduire la quantité d'information en divisant chaque valeur par un coefficient de quantification. Ceci apporte des pertes d'informations de l'image.

Codage
La phase de codage est la seule étape indispensable à tous les algorithmes de compression. Elle permet de générer un flux binaire comprimé. Il existe plusieurs techniques de codage parmi les plus utilisées nous citons les codages : d'Huffman, Arithmétique et EBCOT - (*Embedded Block Coding with Optimized Truncation*).
Dans la pratique, ces 3 phases ne sont pas toujours séparables d'un point de vue algorithmique. La différenciation ici est surtout fonctionnelle.

Chapitre 1

Cryptage

Sommaire :

1.1	Généralités sur le cryptage
1.2	Algorithmes de chiffrement et clefs
1.3	Chiffrement par flot
1.4	Chiffrement par bloc
1.5	Autres cryptosystèmes
1.6	Cryptanalyse

Introduction

La cryptographie est une science très ancienne qui date de 1900 ans avant J.-C.. Des recherches indiquent qu'un scribe égyptien a employé des hiéroglyphes non conformes à la langue pour écrire un message. De ce temps-là et au long de l'histoire, la cryptographie a été utilisée exclusivement à des fins militaires. Aujourd'hui, les réseaux informatiques exigent une phase de cryptographie comme mécanisme fondamental afin d'assurer la confidentialité des information numériques. En se rapportant à des événements historiques de notre ère informatisée [Wil94, New40], nous pouvons retenir les suivants : la machine Énigma créée par Dr Arthur Scherbius en 1923 et qui a été utilisée largement dans la seconde guerre mondiale ; la théorie de l'information de Claude Shannon en 1949 ; l'algorithme *lucifer* développé par IBM en 1970 ; la théorie du système à clef publique par Whitfield Diffie et Martin Hellman en 1976 qui a marqué le début de la cryptographie moderne ; le premier standard pour le cryptage, l'algorithme DES en 1976 ; les résultats de la cryptographie quantique par Charles H. Bennett et Gilles Brassard en 1990, et le standard AES en 2000, qui a remplacé le DES.

1.1 Généralités sur le cryptage

Le cryptage ou (encryptage, chiffrage, chiffrement) peut être défini comme une fonction réversible de transformation des données en envisageant la protection d'information contre toute prise de connaissance du contenu (confidentialité) ou modification indue (intégrité). Le chiffrage a pour but d'assurer l'indéchiffrabilité et l'inintelligibilité des informations.
D'une façon formelle, un cryptosystème est caractérisé par les éléments $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, où $\mathcal{P}$ est l'ensemble des textes clairs possibles, $\mathcal{C}$ l'ensemble des textes chiffrés possibles et $\mathcal{K}$ est l'ensemble des clefs possibles. Pour chaque clef $K \in \mathcal{K}$, il y a une règle de cryptage $e_K \in \mathcal{E}$ et une règle correspondante de décryptage $d_K \in \mathcal{D}$. Chaque $e_K : \mathcal{P} \to \mathcal{C}$ et $d_K : \mathcal{C} \to \mathcal{P}$ sont des fonctions telles que $d_K(e_K(x)) = x$ pour tout texte clair $x \in \mathcal{P}$.

1.1.1 Cryptosystèmes classiques

Cette section est consacrée aux systèmes cryptographiques qui ont été conçus avant la création des ordinateurs et qui ont donné les concepts et les bases pour l'évolution de plusieurs algorithmes symétriques encore utilisés de nos jours. Les cryptosystèmes classiques sont groupés en chiffrement monoalphabétique et polyalphabétique. Le chiffrement **monoalphabétique** est très primaire, il s'agit d'une substitution simple. Chaque lettre est remplacée par une autre lettre ou symbole conformément à l'algorithme [Sti05].

Par décalage (*shift cipher*)

Le principe de ce chiffrement est de décaler les lettres de l'alphabet. Les fonctions de cryptage $e_k(x) = x+K \bmod 26$ et décryptage $d_k(y) = y-K \bmod 26$ sont définies en $\mathbb{Z}_{26}$ et K est la clef de décalage [Sti05]. Par exemple, pour le texte clair $X =$ "$APPELLE$", en remplaçant chaque lettre par celle située 3 cases plus loin ($K = 3$), le texte chiffré "$DSSHOOH$" est obtenu.

Alphabets désordonnés ou codage par substitution

Ce genre de cryptosystème consiste à remplacer une lettre par une autre en utilisant un alphabet désordonné où $\mathcal{K}$ est l'ensemble des permutations aléatoires possibles des 26 lettres (26!). Le chiffrement par décalage est un cas particulier du chiffrement par alphabets désordonnés, où il existe seulement 26 clefs.

Fonctions affines

Il s'agit d'un algorithme qui utilise les fonctions affines du type $e_K(x) = ax + b \bmod 26$ pour le cryptage et $d_K(y) = a^{-1}(y - b) \bmod 26$ pour le décryptage. La constante a est premier avec 26, et a^{-1} désigne l'inverse de a modulo 26. Nous remarquons ici l'évolution de la cryptologie par l'utilisation de l'arithmétique modulaire. Nous remarquons également que si $a = 1$, nous avons le chiffrement par décalage [Sti05].

Blaise de Vigenère a proposé le premier **chiffrement polygraphique**. Dans le chiffrement polygraphique les lettres ne sont pas chiffrées séparément, mais par groupe.

Chiffrement de Vigenère

Le chiffrement de Vigenère [Sti05] utilise une clef qui définit le décalage pour chaque lettre du texte clair. Sa force réside dans l'utilisation non pas de 1, mais de 26 alphabets décalés $(\mathbb{Z}_{26})^m$ où m est un nombre entier positif. Le cryptage $e_K(x_1, \cdots, x_m) = (x_1 + k_1, \cdots, x_m + k_m)$ et décryptage $d_K(y_1, \cdots, y_m) = (y_1 - k_1, \cdots, y_m - k_m)$ sont faits avec les clefs $K = (k_1, \cdots, k_m)$. Le chiffrement de Vigenère est considéré comme un chiffrement par flot, détaillé section 1.3, avec une période m.

Chiffrement de Hill

Le chiffrement de Lester S. Hill [Sti05] est considéré polygrammique. Dans un chiffrement polygrammique un groupe de n lettres est chiffré par un groupe de n symboles. Le principe est de remplacer les lettres par leur rang dans l'alphabet, soit K une matrice inversible, alors $e_K(x) = xK$ et $d_K(y) = yK^{-1}$. En fait, l'idée est de prendre m combinaisons linéaires de m caractères de l'alphabet.

Chiffrement par permutation

Tous les cryptosystèmes présentés jusqu'ici ont évoqué la substitution. D'une autre manière, le fondement du chiffrement par permutation est de garder le même alphabet et de changer seulement l'ordre des lettres. Soit π toutes les permutations $\{1, \ldots, m\}$ pour une clef et m un entier positif, alors les fontions de cryptage et décryptage sont décrites par $e_\pi(x_1, \ldots, x_m) = (x_{\pi(1)}, \ldots, x_{\pi(m)})$ et $d_\pi(y_1, \ldots, y_m) = (y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(m)})$. Le chiffrement par permutation est un cas spécial du chiffrement de Hill pour une matrice de permutation $K_\pi = (k_{i,j})$.

1.1.2 Classification des algorithmes

Les cryptosystèmes peuvent être classifiés conformément à différentes caractéristiques : selon les types de clefs : symétrique, asymétrique ou hybride ; selon les techniques de chiffrement : par bloc ou par flot ; ou selon les corrélations entre le flux de

clefs (*keystream*) et les textes clairs et chiffrés : synchrone ou asynchrone. Le chiffrement symétrique ou à clef secrète, section 1.2.2, est le type de cryptosystème où la clef pour crypter et décrypter est identique. Dans ce cas, l'émetteur et le destinataire partagent cette clef unique. Le chiffrement asymétrique, également appelé chiffrement à clef publique, section 1.2.1, est un système où chaque interlocuteur dispose d'un couple de clefs, la clef publique pour crypter et la clef privée pour décrypter. Le chiffrement hybride, section 1.2.3, fait appel aux deux techniques en même temps, symétrique et asymétrique. Le chiffrement par flot ou par flux, section 1.3, traite les caractères comme une suite de bits, un à la fois, à l'aide d'une transformation qui varie au fur et à mesure du chiffrement [MvOV01]. Les algorithmes de cryptage par flot ont des classifications spéciales. Ils sont considérés comme synchrones, si le flux de clefs est produit indépendamment du texte clair et du texte chiffré. Si le flot de chiffrement est produit à partir de la clef et d'un nombre fixe de caractères du flot chiffré, alors le chiffrement par flot est dit asynchrone.

Le chiffrement par bloc est le type de cryptosystème dont le texte clair est découpé en blocs de tailles fixes et chaque bloc est crypté séparément. Un cryptosystème par bloc peut être symétrique (AES section 1.4.2) ou asymétrique (RSA section 1.2.1) et peut avoir aussi divers modes d'opérations, section 1.4.3.

1.1.3 Notion de sécurité

Dans les années 80, les chercheurs ont formellement défini des notions de sécurité. Il est possible de classer les cryptosystèmes en trois catégories par rapport à la sécurité (parfaite, calculatoire et sémantique).

Sécurité parfaite

La notion de sécurité parfaite ou inconditionnelle a été créée par Shannon [Sha49]. Un cryptosystème est à sécurité parfaite si aucune information peut être obtenue sur le texte clair, à partir du texte crypté correspondant. C'est-à-dire que la probabilité *a posteriori* que le texte clair prenne une valeur x, étant donné que le message chiffré y, soit égale à celle *a priori* que le texte clair prenne la valeur x, $\rho(x \mid y) = \rho(x), \forall x \in \mathcal{P}, \forall y \in \mathcal{C}$. Actuellement, le seul système prouvé inconditionnellement sûr est celui de Vernam, section 1.3.1. En réalité, cette approche est impraticable dans la plupart des scénarios. Diffie et Hellman [DH76] ont proposé le remplacement de la sécurité parfaite par le concept de sécurité calculatoire pour une meilleure évaluation des cryptosystèmes.

Sécurité calculatoire

La sécurité calculatoire est liée à la quantité de ressources informatiques. Elle suppose que si on ne dispose que d'une puissance de calcul limitée, alors il est impossible de déduire le texte clair. C'est-à-dire qu'il est impossible de résoudre certains problèmes de calculs très complexes (la factorisation de grands nombres ou l'extraction de logarithme discret par exemple) dans un temps raisonnable. La sécurité des algorithmes est démontrée en montrant que la capacité d'un adversaire à casser le procédé avec les

ressources existantes et avec une probabilité significative est impossible. Shor [Sho97] a proposé une méthode basée sur un ordinateur quantique (théorique) pour résoudre les problèmes difficiles. Cette méthode suscite que la sécurité calculatoire est un fait directement lié aux avancements technologiques.

Sécurité sémantique

La sécurité sémantique, introduite par Goldwasser et Micali [GM84], s'adresse aux cryptosystèmes asymétriques, et coïncide avec la notion de la sécurité parfaite limitée aux *chiffrements probabilistes*[1]. L'exigence de la sécurité sémantique est très forte et dit qu'on ne peut pas extraire une information, même partielle, sur le texte clair à partir du texte chiffré et de la clef publique. La sécurité sémantique est considérée comme insuffisante parce qu'elle prévoit que l'attaquant a connaissance seulement du texte chiffré et de la clef publique. Elle ne prend pas en compte les autres types d'attaques comme l'attaque à texte chiffré choisi, section 1.6.

1.2 Algorithmes de chiffrement et clefs

1.2.1 Chiffrement asymétrique

La théorie du système asymétrique, publiée par Diffie et Hellman en 1976 a marqué le début de la cryptologie moderne. Ils ont bouleversé la façon de crypter avec l'idée que chaque utilisateur ait deux clefs, une privée et une publique, présentée figure 1.1. Dans leur approche, deux nombres premiers p et g ($g < p$) sont rendus publics. Chaque utilisateur U choisit un numéro secret α tel que $\{\alpha \in \mathbb{Z},\ 0 \leq \alpha \leq p\}$ et calcule $U_\beta = g^{U_\alpha} \mod p$. Si l'utilisateur A (Alice) veut communiquer avec l'utilisateur B (Bob), par exemple, il doit alors choisir son numéro secret A_α, calculer A_β et l'envoyer à B. L'utilisateur B, à son tour, fait la même procédure et envoie B_β à A. Un numéro commun $K = B_\beta^{A_\alpha} \mod p = A_\beta^{B_\alpha} \mod p$ est calculé de chaque côté et utilisé comme clef de cryptage.

La notion fondamentale sur laquelle repose les concepts des principaux algorithmes asymétriques est celle de *fonction à sens unique*[2] à brèches secrètes (*Trapdoor one-way function*). Les brèches secrètes ou trappes sont des informations qui permettent d'inverser facilement les fonctions à sens unique. Pour les algorithmes qui utilisent des fonctions à sens unique, nous citons : le RSA, section 1.2.1, fondé sur la difficulté de factorisation de grands nombres entiers ; l'algorithme de McEliece et l'algorithme de Niederreiter [BL01] établis sur la difficulté du problème de décodage du code linéaire correcteur d'erreur, problème considéré *NP-complet*[3]. Ces deux algorithmes ont été utilisés, récemment, pour la protection de codes des téléphones mobiles [Lou01] ; enfin,

[1] Les chiffrements probabilistes sont ceux qui utilisent des nombres aléatoires. Par conséquence, ceux qui n'en utilisent pas sont appelés déterministes.

[2] Les fonctions à sens unique sont des fonctions faciles à calculer mais infaisables dans un temps réaliste à inverser.

[3] Les problèmes NP-complet sont des problèmes dont la solution ne peut pas être donnée par un algorithme dans un temps polynômial.

l'algorithme de ELGamal a été conçu sur la difficulté à résoudre les problèmes de logarithme discret pour corps fini, section 1.5.1.

Du fait de l'utilisation de grands nombres premiers, les cryptosystèmes asymétriques nécessitent une quantité de calcul importante, ce qui les rend très lents par rapport aux systèmes symétriques. En pratique, ils ne sont pas adaptés au chiffrement d'un volume important de données comme des images. Le RSA, le cryptosystème asymétrique le plus populaire, est 1500 fois plus lent que l'algorithme symétrique DES [Sti05].

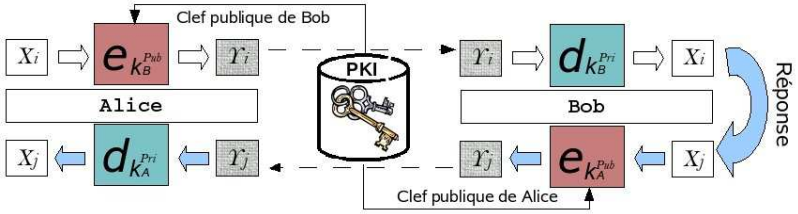


FIG. 1.1 – Principe de chiffrement asymétrique

Théorie des nombres

Des nombreuses fonctions utilisées dans les cryptosystèmes asymétriques viennent des problèmes de la théorie des nombres. L'algorithme RSA, par exemple, utilise : le théorème des restes chinois pour simplifier les calculs d'arithmétique modulaire ; le petit théorème de Fermat pour aider dans l'obtention des fonctions à sens unique à trappe ; l'algorithme d'Euclide pour calculer le PGDC (plus grand diviseur commun) et l'algorithme d'Euclide étendu pour calculer la clef privée [Sti05]. Cependant, Bajard et Imbert [BI04] ont proposé une approche différente pour l'implémentation de l'algorithme RSA. Elle n'est pas basée sur les algorithmes et théorème standards, mais sur le (RNS) *Residue Number System*, version de multiplication de Montgomery.

Théorème des restes chinois

C'est un théorème qui permet de résoudre certains systèmes de congruence. Soit $m_1, ..., m_k$ k entiers positifs deux à deux premiers entre eux et $a_1, \ldots, a_k$ k entiers tels que $0 \leq a_i < n_i$, pour $i = 1, \ldots, k$. Nous calculons alors l'unique entier z tel que $0 \leq z < n$ et $z = a_i \mod n_i$ pour tout $i = 1, ..., k$ où $n = \Pi_i n_i$.

Algorithme d'Euclide étendu

Si d est le plus grand diviseur commun de a et b, il y a donc deux entiers u et v tels que $d = au + bv$. L'algorithme d'Euclide étendu permet de calculer les coefficients u et v.

Petit théorème de Fermat

Soit p un nombre premier et $b \in \mathbb{Z}_p$, alors $b^p \equiv b \pmod{p}$. C'est-à-dire que si un entier b est multiplié par lui-même p fois, et si b lui est soustrait, le résultat est divisible par p.

Algorithme RSA

L'algorithme RSA a été décrit par Ron **R**ivest, Adi **S**hamir et Len **A**dleman [RSA78]. Cet algorithme asymétrique par bloc est très populaire pour le cryptage des données numériques. Le RSA a été breveté en 2000 et fait partie des nombreux protocoles et spécifications. Il constitue le protocole SSL/TLS - (*Secure Socket Layer/Transport Layer Security*) largement utilisé pour l'échange d'information sur internet, section 1.2.3, et forme les spécifications standards des secteurs bancaires et financiers de la France (ETEBAC 5) des EUA (X9.44) et international (SWIFT).

La sécurité du RSA repose sur la difficulté de factorisation de grands nombres entiers. Soit $n = pq$, où p et q sont premiers. $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$ et $\mathcal{K} = (n, p, q, a, b)$. La fonction de cryptage e_K et décryptage d_K sont définies par les équations :

$$y = e_K(x) = x^b \bmod n \qquad et \qquad d_K(y) = y^a \bmod n, \tag{1.1}$$

où $(x, y \in \mathbb{Z}_n)$, $\phi(n) = (p-1)(q-1)$ et $ab \equiv 1 \bmod \phi(n)$. Les valeurs n et b sont publiques et p, q et a sont privées.

Les étapes suivantes font une synthèse de la construction d'un cryptosystème RSA [Sti05].

1. Choisir aléatoirement p et q, deux grands nombres premiers distincts.
2. Calculer $n = pq$ et $\phi(n) = (p-1)(q-1)$.
3. Désigner un entier b, avec $1 < b < \phi(n)$ tel que $pgdc(b, \phi(n)) = 1$
4. Calculer $a = b^{-1} \bmod \phi(n)$ avec l'algorithme Euclidien.
5. Divulguer la clef publique (n, b).

PKI - Infrastructure de gestion de clef

Le chiffrement asymétrique a éliminé le problème de diffusion d'une clef secrète, mais il a posé le problème de la certification de la clef publique. Un utilisateur qui veut chiffrer un message avec un cryptosystème asymétrique nécessite la clef publique du destinataire. Le problème majeur réside dans la nécessité d'associer une clef publique à l'identité de son détenteur légitime. Il est possible de créer une fausse signature numérique en remplaçant la clef publique d'une personne [CdV01]. La certification des clefs a alors un rôle très important dans les systèmes asymétriques.

Il y a deux modèles pour certifier une clef publique. Le premier est fondé sur une relation de confiance directe avec son détenteur. Un exemple, c'est la logique *web of trust* associée à PGP (*Pretty Good Privacy*) [SGS03]. Le second modèle repose sur le fait que tous les interlocuteurs ont confiance en un tiers : les institutions d'infrastructure de gestion des clefs (IGC) en anglais PKI - (*Public Key Infrastructure*). La sécurisation ou authentification de la plupart des clefs publiques est faite à travers le certificat de clef publique fourni par les PKIs. Les normes ISO X509 définissent un modèle très détaillé et très hiérarchisé de ces certificats. C'est-à-dire, la clef d'un utilisateur est certifiée par une autorité dont la clef est à son tour certifiée par une autorité supérieure. Ces normes règlent ainsi les formats de certificats de clefs, qui doivent contenir : l'identité et la clef

publique du porteur du certificat ; la durée de vie du certificat et l'identité de l'autorité de certification qui l'a émis.

Le mécanisme de certification des clefs est très lourd à mettre en œuvre, mais il est fondamental pour sécuriser les systèmes asymétriques de chiffrement. Il y a un nombre important d'institutions PKIs privées dans le monde, particulièrement en Europe, l'institut européen des normes de télécommunication (ETSI) règle leurs créations.

1.2.2 Chiffrement symétrique

Les systèmes de chiffrement symétrique, ainsi appelés chiffrement classique, section 1.1.1, disposent de trois éléments : une clef secrète K, une fonction de cryptage e_K et une fonction de décryptage d_K. Contrairement aux cryptosystèmes asymétriques, les symétriques ont une faible consommation de ressources de calcul et utilisent une clef unique K pour e_K et d_K. Par contre, le désavantage du chiffrement symétrique est l'imposition d'un canal sécurisé pour l'échange de la clef.

La plupart des cryptosystèmes existants sont symétriques. Parmi eux, nous distinguons les algorithmes : one-time pad, A5, COS et RC4/5/6 qui utilisent des techniques de chiffrement par flot, section 1.3, et DES, 3-DES, IDEA, Serpent, Twofish, TEA et AES qui utilisent des techniques de chiffrement par bloc, section 1.4.

Fonctions pseudo-aléatoires

La propriété désirable pour un cryptosystème symétrique, particulièrement ceux par bloc, est que sa sortie soit une suite de bits aléatoires. Néanmoins, une « vraie » suite aléatoire est très difficile à obtenir. John von Neumann signale que de vrais nombres aléatoires sont produits de matériel qui tire parti de certaines propriétés physiques stochastiques, comme le bruit d'une résistance électrique par exemple. Cette difficulté nous impose d'accepter les nombres pseudo-aléatoires, qui sont plus faciles à calculer dans une implémentation informatique et bien acceptés par les cryptologistes et surtout nous pouvons régénérer une séquence pseudo-aléatoire.

Les générateurs de nombres pseudo-aléatoires (GNPA) doivent être rapides, sûrs et rendre leurs sorties uniformément distribuées. Les classes très répandues sont les générateurs basés sur le registre à décalage avec rétroaction linéaire LFSR - (*Linear Feedback Shift Register*). Cette catégorie d'algorithme est très rapide, mais peu sûre. Plusieurs approches ont été proposées pour les améliorer. Coppersmith *et al.* [CKM94] ont proposé l'algorithme *shrinking generator*, qui emploie des combinaisons de LFSRs pour les rendre moins linéaires. Matsumoto et Nishimura [MN98] ont suggéré l'algorithme *Mersenne Twister* fondé sur le TGSFR (*Twisted Generalised Shift Feedback Register*), un type particulier de LFSR. Luby et Rackoff [LR88] ont construit des permutations super pseudo-aléatoires à partir du réseau de Feistel, section 1.4. Blum et Micali [BM84] ont proposé l'algorithme BBS institué sur la permutation à sens unique et sur le problème de *résidu quadratique*[4]. Bruce Schneier et Niels Ferguson ont conçu le GNPA Fortuna,

[4] Un entier naturel q est un résidu quadratique modulo p s'il existe un entier x tel que $x^2 \equiv q \mod p$.

qui est basé sur un algorithme de chiffrement par bloc en mode CTR (*Counter Mode*), section 1.4.3.

Distribution de la clef

Les clefs ont un rôle crucial dans la sécurité des cryptosystèmes [Ker83], et leur distribution est une primitive importante en cryptographie. Dans les systèmes asymétriques les clefs sont authentifiées par les PKIs. D'autre part, les algorithmes symétriques nécessitent un canal sécurisé de communication pour l'obtention de la clef secrète. Il y a deux principales approches d'établissement de clefs en systèmes symétriques : l'**échange de clef** (*key exchange*) où un interlocuteur produit une clef secrète et la transmet à l'autre interlocuteur. Et le **partage de clef** (*key agreement*) où les interlocuteurs s'entendent sur une clef secrète en utilisant une source aléatoire de création.
L'échange de clef nécessite l'existence d'un tiers de confiance TA (*Trusted Authentication Authority*). Le schéma de Blom a été l'initiateur de diverses approches et a stimulé plusieurs travaux récents [CPS03, YG05, SP05]. Blom a proposé un schéma intéressant de pré-distribution de clefs [Blo85]. Dans son schéma, le TA rend public un entier premier p et un élément $r_U \in \mathbb{Z}_!$ pour chaque utilisateur U du réseau. Le tiers de confiance TA choisit ensuite $a, b, c \in \mathbb{Z}$ et construit le polynôme $f(x, y) = a + b(x + y) + cxy \mod p$. Pour chaque utilisateur U, le TA calcule $b_U = b + cr_U \mod p$ et l'envoie sur un canal sûr. Si U et V veulent communiquer entre eux, ils utilisent la clef commune $K = f(r_U, r_V)$. De leur côté U calcule b_U et V calcule b_V. Diffie et Hellman ont proposé un schéma de pré-distribution basé sur leur algorithme asymétrique, présenté section 1.2.1, qui utilise les logarithmes discrets sur un corps fini.

Un schéma très employé pour la distribution de clefs est l'utilisation des protocoles en ligne. Le plus populaire est le protocole **Kerberos**. Son approche consiste à générer des clefs secrètes pour chaque utilisateur dans chaque session. L'utilisateur U reçoit un numéro d'affiliation, lors de la procédure d'enregistrement, et une clef qui est le résultat du chiffrement du numéro d'affiliation. Pour pouvoir correspondre avec un autre affilié V, l'utilisateur U demande à TA une clef de session. Cette clef est générée et envoyée à U et V.

Si nous ne souhaitons pas utiliser un serveur de clefs en ligne, nous devons utiliser une technique de **partage de clefs**. La plus ancienne et la plus connue est la technique de Diffie-Hellman basée sur leur algorithme asymétrique, présentée section 1.2.1. Matsumoto *et al.* [MTI86] ont construit également divers protocoles de partage de clefs, appelés MTI, modifiant les travaux de Diffie et Hellman. Nous trouvons une description détaillée de ces schémas dans l'ouvrage de Douglas Stinson [Sti05].

1.2.3 Chiffrement hybride

Le concept de chiffrement hybride fait appel aux deux techniques, symétrique et asymétrique, comme présenté figure 1.2. Il a été mis en œuvre par Zimmermann pour le PGP (*Pretty Good Privacy*) en 1991 [SGS03]. L'idée est d'utiliser la rapidité de l'algorithme symétrique, et la sécurité de l'asymétrique. Une clef secrète K de 128

bits est générée automatiquement pour la session. Le message m est chiffré avec cette clef K en utilisant un chiffreur symétrique, $m' = e_K(m)$. La clef K est alors chiffrée avec un chiffreur asymétrique en utilisant la clef publique du destinataire B, $K' = e_{k_B^{Pub}}(K)$. Ensuite, le message entier $M = m' + K'$ (message chiffré symétriquement et clef asymétriquement) est envoyé au destinataire. De l'autre côté, B utilise sa clef privée k_B^{Pri} pour décrypter la clef K' et ensuite déchiffrer le message. Un exemple d'un système hybride est le protocole SSL (*Secure Socket Layer*) développé par les sociétés Netscape et RSA Security, cette dernière est responsable de l'algorithme RSA. Dans le chapitre 5, nous proposons une autre approche de chiffrement hybride appliquée aux images en utilisant une insertion de données cachées.

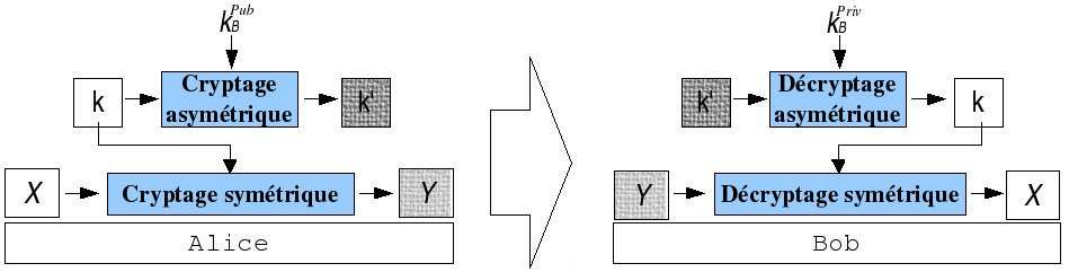


FIG. 1.2 – Cryptage hybride.

1.2.4 Quelques remarques sur les tailles des clefs

Un paramètre essentiel pour la sécurité d'un cryptosystème est la longueur de la clef. Une clef de longueur de 128 bits est considérée comme sûre pour les systèmes symétriques, néanmoins pour les asymétriques cette longueur est considérée comme extrêmement faible. Il existe plusieurs façons d'essayer de casser une clef. La manière la plus simple de trouver une clef de 512 bits, dans le chiffrement RSA par exemple, est d'essayer la factorisation du nombre n, parce que l'attaque exhaustive, détaillée section 1.6, est infaisable (2^{512}). Actuellement, les algorithmes peuvent factoriser des nombres avec 154 chiffres décimaux (512 bits). Cet exploit a été réalisé en 1999 par onze équipe scientifiqueset la factorisation a pris deux mois de calculs répartis sur 300 ordinateurs [CdV01]. De plus, Adir Shamir, un des créateurs du RSA, a proposé une machine théorique qui peut factoriser des nombres premiers de 1024 bits [LTS+03]. De ce fait une clef de 1024 bits n'est plus estimée comme sûre. Aujourd'hui, pour avoir une sécurité équivalente à 128 bits des systèmes symétriques, les systèmes basés sur la factorisation et les systèmes basés sur les logarithmes discrets doivent avoir des clefs de longueur de 2048 bits. Une alternative possible réside dans les cryptosystèmes basés sur les courbes elliptiques qui demandent une clef moins longue pour avoir la même sécurité [LV01]. À titre de comparaison, seulement récemment un système avec une clef de 109 bits basé sur courbe elliptique a été cassée[5].

Au niveau des clefs secrètes des systèmes symétriques, un seul bit ajouté à la taille

[5] http ://openpgp.vie-privee.org/news2000-1.htm

de la clef fait doubler le temps de calcul nécessaire à une attaque exhaustive. Une clef de 65 bits est donc deux fois plus résistante qu'une clef de 64 bits. Aujourd'hui, avec l'évolution des ordinateurs les cryptosystèmes symétriques, de plus en plus, utilisent des clefs entre 128 et 256 bits.

Les normes *NIST FIPS 140-2*[6] indiquent les longueurs des clefs pour les cryptosystèmes les plus utilisés. A titre de comparaison, pour obtenir un niveau de sécurité équivalent à l'algorithme AES avec une clef de 128 bits, l'algorithme RSA doit employer une clef de 3072 bits et le ECC (*Elliptic Curve Cryptosystems*) de 256 bits. Pour une clef de 256 bits en AES, il est nécessaire d'utiliser une clef de 15360 bits en RSA et de 512 bits en ECC.

1.3 Chiffrement par flot

Le chiffrement par flot (*stream cipher*) crypte séparement un caractère individuel (une suite de bits), en utilisant une transformation qui varie au fur et à mesure du temps. Par comparaison, le chiffrement par bloc chiffre un groupe de caractères (bloc) simultanément en employant des transformations fixes sur tous les blocs. L'idée principale du chiffrement par flot est de produire un flux de clefs (*keystream*) $\mathbf{z} = z_1, z_2, \ldots, z_n$ et d'utiliser $\mathbf{z}$, conjointement avec le texte clair $\mathbf{x} = x_1, x_2, \ldots, x_n$ pour générer le texte chiffré $\mathbf{y} = y_1, y_2, \ldots, y_n$.

Le chiffrement par flot est mieux implanté en hardware parce qu'il demande des circuits peu complexes et avec de petites zones mémoire (*buffer*). Il est ainsi conseillé pour les environnements bruités avec des ressources limitées. Ceci est le cas de la téléphonie GSM (*Global System for Mobile Communications*) qui se sert du chiffrement par flot A5/1/2.

1.3.1 Chiffrement de Vernam (One-time pad)

Malgré sa simplicité, le chiffrement de Vernam est la seule méthode sûre de façon inconditionnelle. Celui-ci est impossible à décrypter même avec une puissance de calcul infinie. La fonction de cryptage de Vernam est $y_i = x_i \oplus z_i$, où $\oplus$ est un ou exclusif (XOR), et de décryptage $x_i = y_i \oplus z_i$. Le chiffrement de Vernam est appelé *one-time pad*, quand le flux de clefs $\mathbf{z}$ est généré aléatoirement et indépendamment.

1.3.2 Chiffrement synchrone

Un chiffrement par flot est considéré synchrone, figure 1.3, si le flux de clefs $\mathbf{z}$ est produit indépendamment du texte clair $\mathbf{x}$ et du texte chiffré $\mathbf{y}$. Le chiffrement synchrone peut être décrit par les équations suivantes :

$$\sigma_{i+1} = f(\sigma_i, K), \qquad z_i = g(\sigma_i, K), \qquad y_i = h(z_i, x_i), \tag{1.2}$$

avec σ_0 le vecteur d'initialisation, f la fonction d'état, g une fonction GNPA pour produire le *keystream* $\mathbf{z}$, et h la fonction de sortie qui produit le texte chiffré $\mathbf{y}$. Les

[6] http ://csrc.nist.gov/cryptval/140-1/FIPS1402IG.pdf

éléments du procédé de décryptage sont identiques, il suffit de remplacer $y_i = h(z_i, x_i)$ par $x_i = h^{-1}(z_i, y_i)$. Un exemple classique de chiffrement synchrone sont les modes OFB et CTR des chiffrements par bloc. Le chiffrement synchrone est appelé **chiffrement par flot additif** quand la fonction h utilise un ou exclusif pour le chiffrement $h = (z_i \oplus x_i)$.

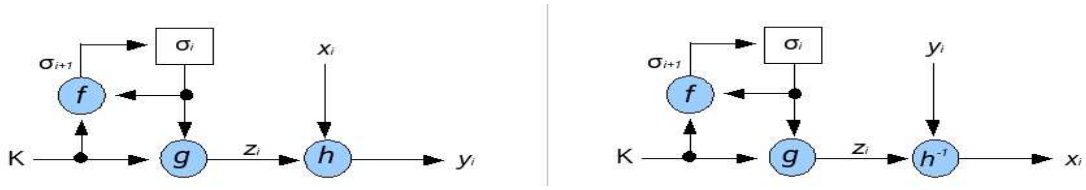


FIG. 1.3 – Cryptage et décryptage synchrone

Propriétés du chiffrement synchrone

Synchronisation : dans un chiffrement synchrone l'émetteur et le récepteur doivent être synchronisés. Si un bit est perdu ou ajouté dans le texte chiffré pendant la transmission, alors **y** ne peut pas être décrypté. Ce fait demande une re-synchronisation qui peut être acquise par une ré-initialisation ou par insertion de marques synchronisantes dans le texte chiffré.

Non propagation d'erreur : si y_i est modifié lors de la transmission, il ne perturbera pas le déchiffrement des textes chiffrés suivants.

Attaques actives : la perte de synchronisation (première propriété) sera immédiatement détectée par le récepteur. Par contre, la deuxième propriété peut passer inaperçue auprès du récepteur. Il est donc important d'introduire des mécanismes garantissants l'authenticité et l'intégrité des données.

1.3.3 Chiffrement asynchrone

La figure 1.4 illustre un système de chiffrement asynchrone (ou auto-synchronisant). Les équations (1.3) montrent que **z** est produit à partir de la clef K et d'un nombre fixe de caractères du flot chiffré **y**. Soit les équations suivantes :

$$\sigma_i = (y_{i-t}, y_{i-t+1}, \ldots, y_{i-1}), \qquad z_i = g(\sigma_i, K), \qquad y_i = h(z_i, x_i), \qquad (1.3)$$

avec $\sigma_0 = (y_{-t}, y_{-t+1}, \ldots, y_{-1})$ l'état initial, K la clef, g la fonction qui produit le flot de clefs, et h la fonction de sortie. Actuellement, le chiffrement auto-synchronisant le plus employé est celui basé sur le chiffrement par bloc en mode CFB en utilisant un bit à la fois [MvOV01].

Propriétés du chiffrement auto-synchronisant

Auto-synchronisation : si des bits sont perdus ou ajoutés dans y_i, le procédé se resynchronise au bout de t textes chiffrés. En effet, le déchiffrement dépend uniquement des t précédents y.

Propagation d'erreur : si y_i est modifié, le déchiffrement des t suivants y sont corrompus.

Diffusion des propriétés statistiques : Du fait que le x_i influe la suite des textes chiffrés

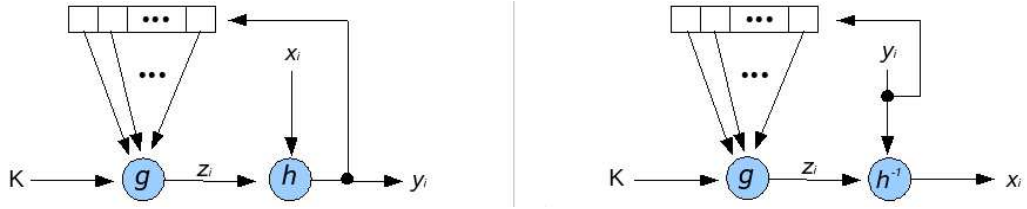


FIG. 1.4 – Cryptage et décryptage asynchrone.

suivants, les propriétés statistiques sont dispersées. Cela rend plus difficile les attaques basées sur une analyse statistique utilisant les redondances du texte clair.
Attaques actives : la modification d'un chiffre se répercute sur les t chiffres suivants, elle a moins de chances de passer inaperçue auprès du récepteur. Par contre, l'ajout ou la suppression de chiffres seront moins détectés que pour les procédés synchrones.

1.4 Chiffrement par bloc

Contrairement au chiffrement par flot qui utilise des transformations qui varient, le chiffrement par bloc utilise des transformations fixes sur tous les blocs de taille également fixe. En fait, le chiffrement par bloc est un cas spécial de chiffrement par flot où $\mathbf{z_i} = K \quad \forall i \geq 1$. Les cryptosystèmes par bloc sont très versatiles et peuvent faire le rôle de GNPA, chiffrement par flot, MAC et fonction d'hachage.

Nous revenons aux théories de Shannon pour citer les propriétés de confusion et diffusion qui sont très liées au chiffrement par bloc. La **confusion** est l'acte de rendre la corrélation entre la clef de chiffrement et le texte chiffré la plus complexe possible. La substitution ou le remplacement d'un symbole par un autre en utilisant une boîte-S (S-BOX) est un mécanisme pour accomplir la confusion. La **diffusion** est l'acte de dissiper la redondance des statistiques du texte clair dans les statistiques du texte chiffré. Elle est associée à la dépendance des bits d'entrée et de sortie. L'effet avalanche, la transposition et le réarrangement de l'ordre des symboles sont des techniques de diffusion.

La plupart des chiffrements symétriques par bloc (DES, 3DES, TEA et IDEA) sont construits sur l'approche du réseau de Feistel.

Réseau de Feistel

Il s'agit d'une construction qui s'appuie sur des principes simples d'opérations répétées de permutations et substitutions des blocs de données, figure 1.5. La clef K est utilisée pour générer une séquence de n sous-clefs qui seront employées dans chaque ronde. Le bloc d'entrée est séparé en deux parties A et B. Une fonction f est appliquée à une des deux moitiés. Une ronde de Feistel calcule A_iB_i à partir de $A_{i-1}B_{i-1}$ selon $A_i = B_{i-1}$ et $B_i = A_{i-1} + f(B_{i-1}, k_i)$. Le résultat est alors combiné avec l'autre moitié à l'aide d'un ou exclusif $\oplus$. L'inversion est très simple et il suffit d'appliquer la même transformation dans l'ordre inverse des sous-clefs [MvOV01].

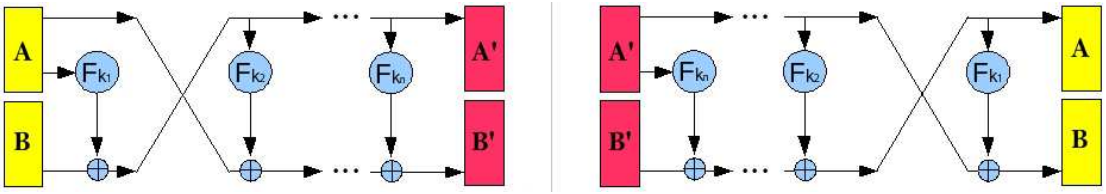


FIG. 1.5 – Cryptage et décryptage du réseaux de Feistel

1.4.1 Algorithme DES et 3-DES

L'algorithme DES a été adopté comme standard en 1976 par le NBS, néanmoins il a été élaboré au début des années 1970 par Horst Feistel. Le DES est un réseau de Feistel à 16 rondes, à clef k de 56 bits diversifiée en 16 clefs de 48 bits, codant des blocs de 64 bits. Il utilise des tables de substitution (boite-S) fixes pour rendre la confusion.

Le Triple DES ou 3DES a été proposé par Walter Tuchman. Il consiste à appliquer les DES trois fois à l'aide de différentes clefs. Plusieurs approches sont possibles en fonction de la quantité de clefs et du type d'opérations [CJM96]. Un chiffrement avec une clef de 112 bits est obtenu par l'application du DES trois fois avec 2 clefs différentes de 56 bits, soit $Y_i = e_{k1}(e_{k2}(e_{k1}(X_i)))$. Le DES-EEE3 a la même approche mais utilise trois clefs distinctes $Y_i = e_{k3}(e_{k2}(e_{k1}(X_i)))$. Le DES-EDE emploie deux clefs pour les opérations cryptage/décryptage/cryptage, $Y_i = e_{k1}(d_{k2}(e_{k1}(X_i)))$. Le triple DES le plus sûr est le DES-EDE3 $Y_i = e_{k3}(d_{k2}(e_{k1}(X_i)))$ [WSB05].

Aujourd'hui, l'ancestral algorithme DES n'est plus recommandé à cause de la longueur trop petite de clef et de sa lenteur d'exécution [DH77].

1.4.2 Standard AES

L'algorithme Rijndael a été conçu par deux chercheurs Belges Joan Daemen et Vincent Rijmen et il a remplacé le DES comme standard AES (Advanced Encryption Standard) en 2000 [DR02b]. L'algorithme AES n'utilise pas la structure de Feistel. Il travaille sur une séquence de transformations, différentes pour la clef et le bloc de données, appliquée n fois (rondes). Le bloc de données et la clef peuvent avoir des longueurs de 128, 192 ou 256 bits, et le nombre de rondes dépend de la combinaison de ces longueurs. Pour un bloc et une clef de 128 bits, par exemple, dix rondes sont employées et les transformations sont appliquées de la façon suivante : d'abord le bloc de données de 128 bits est coupé en 16 octets disposés sous la forme d'une matrice de données X^D de 4×4. Sur la clef, également de 128 bits, est appliquée cette même opération de découpage et réarrangement sous la forme d'une matrice 4×4. L'ensemble des transformations appliquées sur la matrice de données X est présenté figure 1.6. Ces transformations sont employées sur les lignes et colonnes de la matrice de données sur la forme de ronde et chaque ronde consiste en quatre opérations :
AddRoundKey, addition de la clef de ronde : chaque octet de la matrice de données est combiné, à l'aide d'un ou exclusif, avec l'octet correspondant de la sous-clef de ronde. Le premier *AddRoundKey* est fait avec la clef originale.

SubBytes, substitution des octets : c'est une transformation non linéaire des octets. Chaque octet est remplacé à l'aide d'une matrice appelée boîte-S. La boîte-S est un dispositif de confusion et son utilisation a été motivée par les attaques linéaires et différentielles.
ShiftRows, décalage de lignes : il s'agit du décalage cyclique de chaque ligne de la matrice X vers la gauche, selon son numéro de ligne. Ligne 0, aucun décalage ; ligne 1, un décalage ; ligne 2 deux, etc. La motivation a été la résistance contre les attaques différentielles [WLW03].
MixColumns, brouillage des colonnes : c'est une combinaison linéaire où la matrice de données est multipliée par une autre matrice pour réordonner les colonnes. Le *MixColumns* a été utilisé parce qu'il est un mécanisme de diffusion facilement implanté en processeurs de 8 bits.

L'ensemble des transformations appliqués sur la clef est appelé *KeySchedule*. Il est composé de deux composants : l'expansion de la clef et la clef de ronde. Ces transformations, sont appliquées par colonne qui, à son tour, s'appelle mot. Les octets de la dernière colonne sont décalés cycliquement et un mot est ajouté récursivement par un ou exclusif au mot précédent pour créer les clefs de rondes. L'ensemble des transformations sur la clef a pour but d'enlever les symétries et de rendre le chiffrement résistant aux attaques à partir d'un morceau connu de la clef [DR02a].

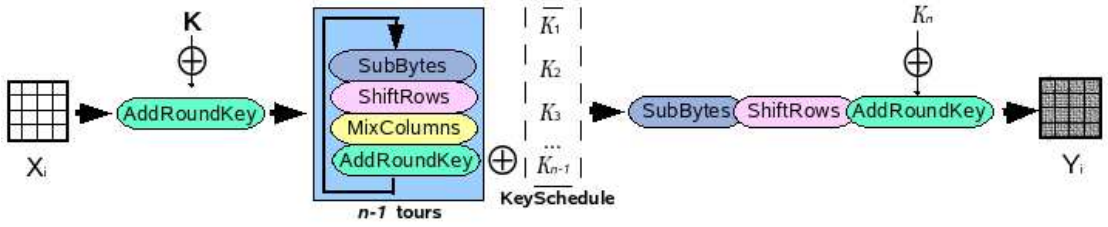


FIG. 1.6 – L'algorithme AES

1.4.3 Modes d'opération

Les algorithmes de chiffrement par bloc ont plusieurs manières de traiter les blocs de texte clair et la clef secrète. Il existe cinq modes standards les ECB, CBC, CFB, OFB et CTR. Cependant, d'autres modes sont actuellement créés afin de faire face aux demandes de sécurité.

Dictionnaire de codes

ECB (*Electronic Code Book*) est le mode standard de chiffrement par bloc. Dans ce mode, tous les blocs sont chiffrés de la même manière. Deux textes clairs $X_i = X_{i+t}$ identiques produiront des textes chiffrés aussi identiques $Y_i = Y_{i+t}$. Ceci est un inconvénient car il devient sensible aux attaques par répétition, présentées section 1.6. Par contre, l'avantage de ce mode est l'indépendance des blocs chiffrés par rapport aux autres, un bloc peut être déchiffré isolément.

Enchaînement des blocs

Dans le mode CBC (*Cipher Block Chaining*), chaque bloc clair X_i est XORé (combiner avec un ou exclusif) avec le bloc précédemment chiffré, $Y_i = e_K(Y_{i-1} \oplus X_i)$. Le tout premier bloc clair est XORé avec un vecteur d'initialisation $VI = Y_0$, qui est généré aléatoirement à l'aide de la clef secrète. Le déchiffrement est fait par $X_i = d_K(Y_i) \oplus Y_{i-1}$. Cet enchaînement rend différemment tous les blocs chiffrés. Le PCB (*Propagating Cipher-Block Chaining*) est une variante du CBC. Il a été conçu pour propager les erreurs pour le contrôle des transmissions. L'idée est d'utiliser deux blocs clairs consécutifs pour le cryptage. Il est particulièrement utilisé dans le protocole Kerberos présenté section 1.2.2. Les fonctions de cryptage et décryptage sont $Y_i = e_K(X_i \oplus X_{i-1} \oplus Y_{i-1})$ et $X_i = d_K(Y_i) \oplus X_{i-1} \oplus Y_{i-1}$.

Chiffrement à rétroaction

Le mode CFB (*Cipher Feedback*) se comporte comme un chiffrement par flot autosynchronisant, présenté section 1.3.3. En fait, tous les modes comportant le terme *feedback* sont considérés par flot, et leur principale caractéristique est la création du flux de clefs (*keystream*). Dans le mode CFB, le flux de clefs z_i est obtenu en cryptant le bloc chiffré précédent y_{i-1}. La première ronde est déclenchée avec le vecteur d'initialisation VI. Le cryptage est $Y_0 = VI$, $z_i = e_K(Y_{i-1})$, $Y_i = z_i \oplus X_i$ et le décryptage est $z_i = e_K(Y_{i-1})$, $X_i = z_i \oplus Y_i$.

Chiffrement à rétroaction de sortie

Le mode OFB (*Output Feedback*) se comporte comme un chiffrement par flot synchrone, présenté section 1.3.2. Le flux de clefs est obtenu en cryptant le précédent flux de clef. Le cryptage est $z_0 = VI$, $z_i = e_K(z_{i-1})$, $Y_i = z_i \oplus X_i$ et le décryptage est $z_i = e_K(z_{i-1})$, $X_i = z_i \oplus Y_i$.

Chiffrement basé sur un compteur

Le mode CTR (*Counter*) a été récemment standardisé (ISO/IEC 10116) et a des caractéristiques très similaires à OFB. Il autorise en plus une propriété d'accès aléatoire pour le décryptage. Il génère la clef dynamique suivante par cryptage de valeurs successives d'un compteur. Le compteur peut se baser sur n'importe quelle fonction GNPA. On utilise $z_0 = ctr$, $z_i = e_K(ctr + i)$ et $Y_i = z_i \oplus X_i$.

Nous distinguons également un mode de chiffrement très répandus aujourd'hui, le **AE** (*authenticated-encryption*). Il s'agit d'une classe d'algorithmes de chiffrement par bloc qui a comme but de chiffrer et d'authentifier le message simultanément. L'idée est d'ajouter des informations spécifiques dans les blocs pour l'authentification [Rog02]. Ce principe a été déclenché en 1985 à travers l'algorithme CBC-MAC *message authentication code* qui utilise le mode CBC et un code d'authentification rajouté dans le dernier bloc chiffré. Le principe du CBC-MAC a fait ressortir d'autres algorithmes comme : le CCM (*Counter-Mode/CBC-MAC*), créé par *RSA Laboratories*, qui utilise l'algorithme AES en mode CTR. Cet algorithme a été adopté comme le standard du protocole IEEE 802.11 pour la communication WiFi ; le EAX (*Two-Pass Authenticated-Encryption Scheme*) proposé par Bellare *et al.* [BRW04] pour résoudre le problème de

authentification AEAD *authenticated-encryption with associated-data*, et qui est entrain de remplacer le standard CCM.

Chaque mode décrit a différents avantages et inconvénients. Dans les modes ECB, OFB et CTR, par exemple, tout changement dans le bloc du texte clair X_i provoque une modification dans le bloc chiffré correspondant Y_i, mais les autres blocs chiffrés ne sont pas affectés.

1.5 Autres cryptosystèmes

Nous avons décrit, section 1.2.1, les algorithmes basés sur la difficulté de factoriser de grands nombres premiers. Néanmoins, il existe d'autres approches mathématiques fondées sur la difficulté d'autres catégories de problèmes et aussi bien que des approches basées sur les lois de la physique qui peuvent être utilisées pour construire des algorithmes de cryptage. Parmi ces nombreuses approches nous allons en citer quelques unes à titre d'exemple.

1.5.1 Approches mathématiques

Certains cryptosystèmes sont basés sur le problème du **logarithme discret** DLP (*Discrete logarithm problem*). Il s'agit de trouver l'unique entier $a \in \mathbb{Z}_p$ tel que $\alpha^a \equiv \beta \pmod{p}$ où p est premier et $a = log_{\alpha}^{\beta}$. Le calcul des logarithmes discrets est difficile, et il n'existe pas d'algorithmes efficaces pour le résoudre, si ce n'est d'utiliser le problème inverse de l'élévation à une puissance discrète. Des schémas basés sur le DLP les plus célèbres sont le cryptosystème asymétrique de ElGamal [Sti05] et le standard américain DSA *Digital Signature Algorithm* pour la signature numérique [JM99].
Les systèmes utilisant des **courbes elliptiques** sont basés sur la difficulté de résoudre le problème ECDLP (*Elliptic Curve Discrete Logarithm Problem*). Une courbe elliptique E sur un corps fini $\mathbb{Z}_p$ (peut être définie aussi sur $GF(q)$ ou $GF(2^k)$) est une courbe définie par l'équation simplifiée $Y^2 = X^3 + aX + b \pmod{p}$ où $(a, b \in K)$ et p un premier tel que $p > 3$. Le problème consiste à, à partir de deux points sur la courbe (P et $Q \in \mathsf{E}(\mathbb{Z}_p)$, de trouver un nombre k tel que $Pk = Q$. Le nombre k est appelé logarithme discret de Q en base P. Les ECCs (*Elliptic curve cryptosystem*) utilisent des clefs plus courtes que les autres systèmes asymétriques, par contre les opérations de chiffrement et de déchiffrement sont plus complexes [BINP03]. Cependant, les ECCs peuvent être également utilisées pour la factorisation de grands nombres entiers [LJ87]. Ceci est considéré comme une menace au RSA face à la montée en puissance des processeurs. Malgré le début de l'utilisation des courbes elliptiques en cryptographie depuis plus de vingt ans (1985 par Neal Koblitz et Victor Miller), celles-ci ne sont pas très connues du fait qu'un grand nombre de brevets empêchent leurs développements. Deux algorithmes très connus basés sur les courbes elliptiques sont : le cryptosystème asymétrique l'EC-ElGamal [Sti05] et l'algorithme d'échange de clefs EC- Diffie et Hellman [Sti05].

Les **codes correcteurs d'erreurs** sont fondés sur des problèmes de la théorie des codes, et essentiellement sur la difficulté du problème du décodage. Il s'agit en fait de

décoder et/ou identifier un code dont la structure ou les paramètres sont cachés. Ce genre de cryptosystème n'est pas bien accepté à cause des clefs qui sont de grandes matrices. Nous distinguons en particulier l'algorithme de McEliece [McE78].

Pour terminer avec les approches mathématiques nous faisons référence à deux classes de problèmes qui ont été utilisées pour la cryptographie asymétrique : les systèmes fondés sur les **problèmes combinatoires** et sur les **problèmes algébriques** abordés dans les travaux de thèse de Perret [Per05].

1.5.2 Approches physiques

Le cryptage chaotique et le cryptage quantique sont des exemples d'emploi des propriétés physiques pour le chiffrement d'information.
Dans le **cryptage chaotique**, le chiffrement d'un message s'effectue en superposant à l'information initiale un signal chaotique produit par un générateur de chaos optique par exemple. Une équipe de chercheurs français (OCCULT) a utilisé une onde porteuse de message générée par un laser semi-conducteur opérant en mode chaotique à travers un réseau de 120 km de fibre optique[7].

Le **cryptage quantique** profite des propriétés physiques des particules de la lumière (photon). La polarisation du photon et son angle de vibration peuvent être utilisés pour quantifier de façon chiffrée les informations. La physique quantique peut être utilisée ainsi pour produire des bruits aléatoires et générer des vrais nombres aléatoires très désirés en cryptographie.

1.6 Cryptanalyse

La cryptanalyse ou l'attaque regroupe tous les moyens de déchiffrer un texte crypté sans avoir connaissance de la clef. Les procédés de cryptanalyse pour le chiffrement symétrique sont très nombreux et la plupart des attaques sont spécifiquement adaptées aux techniques de chiffrement. On distingue les différents types d'attaques en fonction des données supposées connues par les attaquants.

- L'**attaque à texte chiffré seulement** (*Ciphertext-only attack*) : l'attaquant a connaissance du texte chiffré de plusieurs messages.
- **L'attaque à texte clair connu** (*Known-plaintext attack*) : le cryptanalyste a accès à plusieurs textes chiffrés ainsi qu'aux textes clairs correspondants.
- **L'attaque à texte clair choisi** (*Chosen-plaintext attack*) : l'attaquant a accès à l'algorithme de chiffrement. Il l'utilise pour générer des couples (X_i, Y_i) de son choix. La différence principale par rapport l'attaque à texte clair connu est que le cryptanalyste peut choisir le texte à chiffrer.
- **L'attaque à texte chiffré choisi** (*Adaptative-plaintext attack*) : le cryptanalyste a accès à l'algorithme de décryptage. Il peut choisir les textes à déchiffrer sans connaître la clef.

[7] http ://www.futura-sciences.com/news-occult-cryptographie-basee-chaos_7753.php

- **L'attaque par force brute** (*Brute-force attack*) ou l'attaque exhaustive :l'attaquant essaie toutes combinaisons de clefs possibles jusqu'à l'obtention du texte clair.
- **L'attaque par canaux auxiliaires** : toutes les façons d'analyser les propriétés inattendues d'un algorithme sont prises en compte pour réussir à casser le cryptosystème. Dans les algorithmes de chiffrement implémentés en hardware, par exemple, la consommation électrique pour chaque type de calcul du chiffrement peut être utile pour déduire certaines informations de la clef.

La création de techniques modernes de chiffrement a fait ressortir des nouvelles méthodes de cryptanalyse. Le souci le plus grand des cryptographes alors est le classement de ces schémas cryptographiques selon leur niveau de sécurité face aux attaquants. Nous pouvons grouper les diverses techniques de cryptanalyse en deux grandes familles.

Cryptanalyse différentielle

Elle a été proposée par Eli Biham et Adi Shamir en 1991. Elle permet de trouver la clef en utilisant une quantité de textes clairs. L'idée est de fournir comme entrée des textes clairs avec de légères différences (un bit par exemple). Ensuite, on analyse statistiquement le comportement des sorties selon les entrées pour retrouver la clef. En regardant comment les différences en entrée affectent les sorties, on peut établir des règles statistiques. Il existe plusieurs variantes des cryptanalyses différentielles, nous distinguons : différentielle tronquée, différentielle d'ordre supérieur et différentielles impossibles.

Cryptanalyse linéaire

Elle a été inventée par Mitsuru Matsui en 1993 [Sti05]. Elle nécessite une quantité n de couples (texte clair, texte chiffré), tous chiffrés avec la même clef. Le principe est que le même message soit chiffré plusieurs fois avec des clefs différentes pour construire une immense table (téraoctet) qui contient toutes les versions chiffrées de ce message. Lors d'une interception d'un message chiffré, on peut le retrouver dans la table et obtenir la clef qui avait été utilisée pour le cryptage. Cette attaque n'est bien sûr pas faisable car nous aurions besoin d'une table trop importante. Le génie d'Hellman a été de trouver un moyen pour réduire cette table, processus réalisable. Celui-ci consiste à faire une approximation linéaire de l'algorithme pour le simplifier.

Conclusion

Dans ce chapitre, nous avons présenté plusieurs techniques et quelques théories de la cryptographie qui vont nous permettre de comprendre cet axe de recherche très important pour la sécurisation d'information. Nous avons tout d'abord évoqué les notions formelles de sécurité et leurs implications. Nous avons ensuite abordé les différents types de classifications des algorithmes de chiffrement et leurs contextes d'applications. Nous avons observé que les clefs ont un rôle important et qu'une définition correcte de leur longueur est cruciale pour rendre sûrs les cryptosystèmes. Ce chapitre a introduit aussi

les principaux algorithmes de cryptage symétrique, asymétrique, par flot et par bloc, et a présenté également les différentes formes d'attaques et leurs classifications.

Il n'existe pas un standard pour le cryptage d'images. En fait, d'un point de vue pratique et commercial, il est très difficile de l'avoir à cause des particularités de nombreux formats de compression d'images. Nous verrons dans la section 4.2 qu'il existe des avantages et des inconvénients dans l'utilisation des algorithmes pour le cryptage d'images. Nous avons utilisé : l'algorithme RSA pour le cryptage asymétrique et les propriétés du chiffrement asynchrone pour proposer une nouvelle méthode de cryptage résistante au bruit (chapitre 4). Nous utilisons également l'algorithme AES en mode CFB pour le cryptage par flot dans le chapitre 5.

La cryptologie et la cryptanalyse ont été séparées et la cryptanalyse (section 1.6) est devenue une science à elle toute seule depuis les années 1970. Les cryptographes construisent les schémas de chiffrement et les cryptanalystes essayent de les casser. La preuve de sécurité d'un schéma de chiffrement est très complexe. S'il existe une attaque contre le schéma proposé, on sera alors capable de résoudre un problème difficile au sens de la théorie de la complexité. Une telle preuve est appelée une preuve de sécurité [PHA05].

Dans cette thèse, nous prenons le rôle de cryptographes et donc nous ne traitons que superficiellement l'aspect cryptanalyse.

Chapitre 2

Insertion de données cachées

Sommaire :

2.1	Généralités sur l'IDC
2.2	Stéganographie
2.3	Marquage d'image fixe

Introduction

La cryptographie a été une première proposition pour sécuriser des transferts de documents numériques. Aujourd'hui les algorithmes de chiffrement modernes, avec des clefs de longueur importante, permettent d'assurer la confidentialité. Néanmoins, une fois décrypté, le document n'est plus protégé et il peut être distribué ou modifié malhonnêtement. La dissimulation d'information plus particulièrement l'insertion de données cachées peut être une réponse à ce problème. L'insertion d'une marque dans un document permet de l'authentifier et de garantir son intégrité.

La figure 2.1 présente plusieurs techniques de dissimulation d'information. Ces techniques disposent de différentes caractéristiques et sont classifiées selon leurs applications et objectifs.

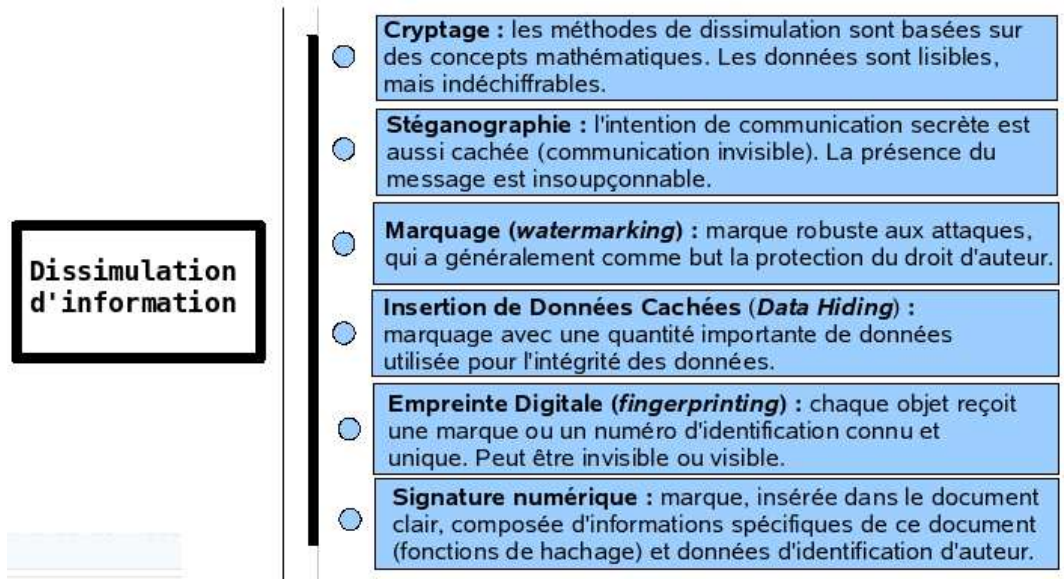


FIG. 2.1 – Techniques de dissimulation d'information.

Dans la dissimulation d'information, le problème classique pour la communication avec des données cachées a été proposé premièrement par Simmons [Sim83]. *Alice* et *Bob* ont été emprisonnés dans des cellules distinctes. Ils désirent développer une stratégie pour leur évasion, mais toute communication entre eux doit passer par la guichetière *Wendy* qui ne permet pas de communication soupçonneuse. La manière de communiquer secrètement est de cacher les informations significatives dans des messages inoffensifs.

La figure 2.2 présente les termes particuliers aux techniques d'insertion de données cachées (IDC), qui seront utilisées tout au long de ce document. La **Couverture** est le fichier original qui va être utilisé pour couvrir ou cacher une information. Le **Message** ou la **marque** est l'information à cacher. Le **Porteur** est le fichier qui porte un message encapsulé. Plusieurs supports numériques peuvent être utilisés comme porteur pour cacher des informations comme par exemple les protocoles de communication [Sin04, SSM03] et les cellules de mémoire [FGS05]. Cependant, le support numérique le plus

FIG. 2.2 – Problème classique de communication secrète.

employé est le fichier sous toutes ses formes, particulièrement celui sous la forme de données multimédias (documents texte, son, image, vidéo).

Dans les sections suivantes, nous présentons quelques théories et aspects des nouvelles techniques d'IDC, stéganographie et marquage d'image.

2.1 Généralités sur l'IDC

2.1.1 Conditions requises

Les méthodes d'insertion de données cachées demandent différentes propriétés selon leurs domaines d'application et leurs finalités.

Indétectabilité

L'indétectabilité est la capacité d'un message à ne pas être détecté par des analyses statistiques. Cette condition est indispensable pour les systèmes de communication secrète basés sur la stéganographie.

Invisibilité/imperceptibilité

Les données cachées doivent être entièrement invisibles par le système visuel humain (SVH). L'opération d'insertion ne doit pas détériorer le porteur d'une façon perceptible. L'invisibilité est une propriété fortement liée au marquage invisible et à la stéganographie.

Spécificité

Le message caché, après certains types d'attaques, peut subir des distorsions et devenir illisible. Le message doit être donc suffisamment spécifique pour être clairement identifiable lors de son extraction. Cette caractéristique est liée au marquage non-aveugle et marquage semi-aveugle, section 2.3.1.

Robustesse

La robustesse est l'aptitude à préserver les données cachées (message) face aux attaques. La robustesse correspond donc à la quantité d'énergie que possède la marque insérée. Une marque de forte énergie est robuste. Cette demande est fortement attachée à la plupart des types de marquages, particulièrement ceux pour la protection des droits d'auteurs.

Capacité

La capacité est la quantité d'information que le fichier couverture peut dissimuler. Elle est généralement mesurée en bits. Dans le contexte de marquage pour la protection des droits d'auteurs, la capacité n'est pas primordiale. L'insertion d'un numéro d'identification codé sur 64 bits suffit dans la plupart des applications. Néanmoins pour la stéganographie et l'IDC - (*data hiding*), cette propriété est très importante.

Détection et extraction

La **détection** est la découverte de la présence d'un message caché m, sans en connaître son contenu. Il existe donc une probabilité P_{ed} d'erreur de ne pas détecter ce message, même s'il est présent. D'autre part, il existe aussi une probabilité P_{fa} de fausse alarme, laquelle est la probabilité de détecter la présence d'un message qui n'existe pas. Dans la détection, le souci est de trouver le meilleur arrangement de la probabilité d'erreur et de la probabilité de fausse alarme. Une forte probabilité P_{ed} pour les schémas de gestion de propriété n'est pas tolérée, néanmoins pour les schémas stéganographiques peut être acceptable. Une forte probabilité P_{fa} n'est, en général, pas souhaitable dans les deux cas à cause du chargement du système (essais et échecs d'extraction, re-analyses, etc).

L'**extraction** est l'obtention du message m à partir du fichier porteur I_p. L'extraction du message est interprétée différemment de la détection. Dans l'extraction, nous considérons toujours que l'image est marquée. Nous prenons en compte la probabilité P_{ee} d'erreur d'extraction (extraction d'un message incorrect). Cette probabilité est très liée à la capacité du message. Plus la marque est invisible et robuste moins l'extraction sera performante et plus grande sera la probabilité d'erreur. Il existe une catégorie de marquage dont la probabilité P_{ee} est très faible, il s'agit du marquage non-aveugle où la couverture I_c est demandée lors de l'extraction.

2.1.2 Domaines d'insertion

Les techniques d'IDC appliquées aux images sont liées aux différents espaces de représentations appelés domaines, et chaque domaine d'insertion dispose de divers schémas de marquage.

Spatial

Dans ce domaine, les méthodes modifient directement la valeur de la couleur du pixel. Ce sont des méthodes simples et peu coûteuses en temps de calcul. Elles sont consacrées aux marquages en temps réel demandés dans des environnements de faible puissance. Certaines techniques dans le domaine spatial peuvent être robustes aux attaques de type transformations géométriques, présentées section 2.3.3.

Fréquentiel

Des schémas de marquage peuvent effectuer l'insertion du message dans des espaces transformés. Un espace transformé est obtenu après l'emploi d'une DCT - Transformée

en Cosinus Discrète ou DFT - Transformée de Fourier Discrète. Cette stratégie rend le message plus robuste à la compression, puisqu'elle utilise le même espace qui sert au codage de l'image. Contrairement au domaine spatial, la marque insérée dans le domaine fréquentiel est très sensible aux transformations géométriques parce que ce genre de transformations modifie considérablement les valeurs des coefficients transformés.

Multi-résolution

L'espace multirésolution pour les images est devenu populaire après la création de la norme JPEG2000, décrite chapitre 3, qui utilise la DWT - Transformée par ondelettes discrète pour transformer le domaine. L'image est décomposée en sous-bandes, ceci permet un isolement affiné des composantes basse-fréquences, qui est un espace d'insertion moins sensible. En plus, le contenu spatial de l'image est aussi conservé et peut être utilisé pour l'insertion d'information. D'autre part, la DWT est considérée comme une décomposition en canaux perceptifs qui facilite l'utilisation d'un modèle psychovisuel.

2.1.3 Mesures et modèles perceptueles

Dans les systèmes d'insertion de données cachées, la mesure de la perturbation apportée sur l'image lors de l'insertion du message est très importante. La démarche la plus employée est alors d'utiliser une métrique d'erreur quadratique moyenne (EQM) pour calculer le PSNR - (*Peak Signal to Noise Ratio*).

PSNR

Le PSNR est la mesure de la distorsion entre le signal marqué et le signal original. Il est défini par :

$$PSNR = 10\log_{10}\left(\frac{max^2}{EQM}\right) \qquad , \qquad EQM = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}||I_c(i,j) - I_p(i,j)||^2$$

où I_c est l'image couverture, I_p l'image porteuse, m le nombre de lignes, n le nombre de colonnes et max la dynamique du signal (pour les images la valeur maximale du pixel codé sur un octet est $max = 255$). L'unité du PSNR est le décibel dB, plus il est élevé et moins la distorsion est importante.

Malgré l'utilisation courante du PSNR pour mesurer la qualité des images, celui-ci n'est pas bien ajusté au Système Visuel Humain - SVH. Le SVH ne perçoit pas tous les signaux de la même façon, comme la sensibilité au contraste par exemple. L'utilisation seule du PSNR ne peut donc pas être considérée comme une mesure objective de la qualité visuelle d'une image [WBL02].

La figure 2.3 montre que, même si le PSNR de l'image (b) est inférieur à celui de l'image (c), l'image (b) possède une meilleure qualité visuelle.

D'autres métriques à prendre en compte sont les phénomènes de perception humaine. Des méthodes de mesures perceptuelles des distorsions qui prend en compte des elements psychologiques et physiologiques de la perception humaine pour l'évaluation

(a) (b) (c)

FIG. 2.3 – Évaluation du PSNR comme mesure de qualité visuelle. (a) Image originale (b) Image fortement contrastée PSNR=24,63 dB, (c) Image fortement comprimée PSNR=28,80 dB.

qualitative des images tatouées [NMC+06, AGB03]. Néanmoins, l'évaluation qualitative est encore d'actualité et il n'y a pas encore une métrique standard. Les approches les plus connues sont les métriques avec **pondération perceptuelle** et par **seuil de perception** présentées ci-dessous.

Pondération perceptuelle

L'approche la plus pratique est l'introduction d'une pondération perceptuelle w au sein de la mesure classique d'erreur quadratique moyenne. Le wPSNR - (*weighted PSNR*) est défini par

$$wPSNR = 10\log_{10}\left(\frac{max^2}{wEQM}\right) \qquad , \qquad wEQM = \frac{1}{n}\sum_{i=1}^{n}\varphi_i^2.(x_i - y_i)^2,$$

où φ_i est une pondération représentant l'importance du i^{eme} échantillon. Plusieurs pondérations ont été proposées [WBL02, Wat93]. La pondération la plus connue est celle de Watson.

$$\varphi_i^2 = \frac{1}{\sigma_{b_i}^2 + V_i^2} \qquad , \qquad V_i = \frac{1}{\|\phi_i\|}\sum_{j\in\phi_i}|x_j|^\rho.$$

Dans ces formules, pour le i^{eme} coefficient, V_i est une mesure d'activité de voisinage, ϕ_i (de taille $\|\phi_i\|$) est l'ensemble qui représente les indices des voisins et enfin la variable σ_{b_i} est un seuil de visibilité qui dépend de la distance d'observation. Ce seuil est fixé à 10^{-2} pour JPEG2000. Les meilleurs résultats sont obtenus pour $\rho = 1/2$. Le wPSNR de Watson a été conçu pour les images dans le domaine DCT notamment JPEG. Il utilise une table de niveau de sensibilité pour les 64 coefficients d'un bloc DCT. Ceci permet de prendre en compte la sensibilité fréquentielle et les phénomènes de masquage dûs à la luminance et au contraste. Une version plus simplifiée du wPSNR de Watson est utilisée dans JPEG2000.

Seuils de perception

Contrairement aux critères de qualité sous forme de pondérations vus précédemment, ce type de seuil ne permet pas de quantifier la distorsion perceptuelle introduite. Néanmoins, il indique le niveau de distorsion maximal JND (*just noticeable difference*) acceptable afin que le changement sur l'image ne soit pas visible. Au-dessous de ce seuil, la modification ne pourra pas être perçue, mais au-dessus elle pourra être remarquée. Les seuils de perception imposent des problèmes de contraintes, rester au-dessous de ce seuil ne permet donc pas d'introduire une marque de forte énergie, et la robustesse est une propriété primordiale pour la plupart des types de marquage. L'utilisation de seuils de perception rend difficile la conciliation entre distorsion perceptuelle et énergie insérée. Watson *et al.* [WYSV97] a précisé expérimentalement des seuils de perception du bruit pour les coefficients DWT.

2.2 Stéganographie

La stéganographie consiste à dissimuler des données dans d'autres données de sorte que l'intention de communication secrète soit également cachée. La stéganographie a accompli un rôle important dans l'histoire de dissimulation d'information [New40, Wil94]. Les premiers emplois de stéganographie sont relatés par Hérodote [Kah92]. Plus récemment, on pense que les terroristes des attentats de 11 novembre 2001, se servaient de la stéganographie pour envoyer aux États-Unis leurs tactiques pour les attentats. Ils ont dissimulé leurs stratégies dans des images pornographiques et les ont envoyé par Internet [Sie01].

2.2.1 Classifications des techniques de stéganographie

Plusieurs approches sont possibles pour classifier les techniques de stéganographie. Celles-ci sont fonction du type de couvertures, du type d'algorithmes ou du schéma d'insertion. Katzenbeisser et Petitcolas [KP00] ont classifié les techniques de stéganographie de la manière suivante :

Substitution

Les méthodes classées dans ce groupe remplacent les parties redondantes de la couverture par le message. Les algorithmes sont simples à mettre en œuvre, mais sont vulnérables à des modifications les plus simples. Parmi les nombreuses méthodes de substitution, nous citons le **remplacement du LSB-1**, bit de poids le plus faible, par le bit du message [BGM96, JJ98]. Rodrigues *et al.* ont proposé le SB4 une méthode moins vulnérable, en **remplaçant le LSB-4** par le bit du message [RRP04]. Une autre approche est le calcul de la **parité d'une région**. L'image est divisée en régions disjointes et chaque région est utilisée pour cacher un bit du message. Il suffit de changer le LSB d'un pixel quelconque de la région afin que la parité soit conforme au bit du message. Les images basées sur **palette de couleurs** sont également utilisées pour l'insertion d'information. Les techniques les plus utilisées sont celles fondées

sur le changement de l'ordre ou d'échelle des couleurs dans la palette [Fri99] et celles basées sur la modification du LSB du vecteur de couleurs [NKE02]. Une autre méthode de substitution est l'utilisation des **erreurs d'arrondis** produites par les procédés de quantification et de *dithering* dans l'image [WBSH05, LG05].

Transformation de domaines

Les procédés insèrent le message dans des espaces transformés comme DCT, DFT, ou DWT, par exemple. Cette stratégie rend le message plus robuste aux attaques parce que l'information est dissimulée dans la portion plus significative du signal. La DCT appliquée aux images est devenue très populaire après la création de la norme JPEG présentée chapitre 3. Par conséquence, beaucoup de stratégies d'insertion de données cachées ont été développées pour ce format d'image. Des méthodes proposent de modifier des valeurs de la table de quantification standard du JPEG [TPC05, CCC02, TC04b]. D'autres méthodes suscitent l'altération des coefficients AC et DC de la DCT [TL03, SJC+02, CCH05]. Aujourd'hui, à cause de la norme JEPG2000 la DWT est devenue la vedette des transformations de domaines pour les images. Ceci a fait surgir diverses techniques qui utilisent des ondelettes pour dissimuler l'information [NSSK02, KMK03, KH05].

Distorsion

L'idée est de cacher le message en effectuant une distorsion du signal original. L'extraction du message est faite par la différence entre l'image porteuse et la couverture. En fait, cette catégorie n'est pas adaptée à la stéganographie à cause de la quantité importante de distorsions apportées à l'image et la nécessité d'avoir l'image originale lors de l'extraction [AI04].

Étalement de spectre

Les algorithmes dans ce groupe adoptent des techniques de modulation de données existantes dans le domaine de communication *spread spectrum*. Le signal original occupe une bande de fréquences beaucoup plus large que la bande minimale nécessaire, ceci afin de combattre les signaux interférant et les distorsions liées à la propagation [KP00]. Les méthodes qui appliquent cette technique sont plus robustes aux attaques parce que le message caché (signal de faible énergie et de très haute fréquence) est étalé sur une bande de fréquences plus large. Nous citons les travaux suivants [WAW00, TC04a, SMCM05].

Modification statistique

Le message est inséré par le changement des propriétés statistiques du fichier couverture. L'extraction est faite par des tests d'hypothèses. L'idée est de diviser l'image originale en régions disjointes et certaines propriétés statistiques de chaque région sont calculées. Si le bit à insérer est 1 la propriété est modifiée, sinon elle n'est pas altérée. L'application pratique de ce type de méthode est très incertaine, parce que la détermination des propriétés statistiques n'est pas élémentaire. Les algorithmes sont spécifiques vis-à-vis de la propriété explorée [SSM+05, Pro01, DW05].

A titre de contribution nous signalons l'existence de la technique de **génération de couverture**, il s'agit de construire une image quelconque, pour faire le rôle de couverture lors d'insertion du message. Cette technique produit des images synthétiques et surréalistes qui ne sont pas bien adaptées à la stéganographie.

Avant de passer à la section suivante, nous devons faire quelques considérations. L'objectif premier de toutes ces techniques de stéganographie est de permettre d'insérer une quantité importante d'informations sans prendre en compte la robustesse. Nous signalons ainsi que la propriété cruciale pour la stéganographie est l'invisibilité et que les schémas d'insertion et d'extraction sont fortement attachés à la clef et à l'algorithme d'insertion.

2.3 Marquage d'image fixe

La puissance des ordinateurs et les réseaux publics ont facilité l'accès et la modification des informations. Malheureusement, la délinquance numérique a fortement augmenté le piratage. La protection des droits d'auteur est devenue donc un enjeu majeur. L'insertion d'une marque identificatrice dans les images est une solution possible.

Le domaine du marquage des données étant une science jeune, il reste encore de nombreux travaux à mener pour la maîtriser complètement [DP03]. Les différents auteurs emploient différentes significations pour le mot *Watermarking*, aquamarquage, tatouage ou simplement marquage. Néanmoins, il y a un consensus : il s'agit d'une marque, insérée dans des fichiers numériques, qui a comme objectif l'identification. Les conditions que doit remplir cette marque dépendent du problème à traiter et les algorithmes doivent prendre en compte les spécificités des images comme la couleur, la résolution et les normes de compression [NP99].

2.3.1 Classification des techniques de marquage

Il est possible de grouper les techniques de marquage selon différentes classifications : conformément au type de clef appliquée (asymétrique et symétrique) ; selon l'information nécessaire à l'extraction (aveugle, semi-aveugle et non-aveugle) ; conformément à la robustesse (fragile, semi-fragile et robuste) ; quant à la perception du SVH (visible et invisible) ; selon la préservation de l'image originale (inversible et non-inversible) et conformément à la technique d'insertion (additive et substitutive). Davoine *et al.* [DP03] et Cox *et al.* [CMB02] ont bien détaillé dans leurs livres ces types de classifications.

Aveugle, Semi-aveugle et Non aveugle

Le marquage **aveugle** est la plus ancienne forme de tatouage. Il n'oblige pas l'extracteur d'avoir connaissance de l'image originale, ni de la marque. Seule l'image porteuse et la clef secrète doivent être disponibles au moment de l'extraction. Dans cette catégorie de marquage on peut distinguer les travaux [YLLS00, CCGN03].

Dans le cadre d'un système **semi-aveugle**, nous avons besoin d'informations supplémentaires pour aider la détection ou l'extraction. Cette demande est dûe à la perte de

synchronisation à cause de canal bruité ou de la technique d'insertion. Lors de l'extraction peut être requise la marque ou la porteuse originale (l'image originale juste après l'incrustation de la marque). Nous remarquons les travaux [SkCjJyEy03, EE06, LVM04].
Au contraire du marquage aveugle, les algorithmes de marquage **non-aveugle** nécessitent toujours l'image originale [GTOMD05, SZT04]. Le nombre d'algorithme non-aveugle n'est pas important par rapport aux nombreux algorithmes semi-aveugles et aveugles. Ceci est parce que la disponibilité des données originales au moment du décodage, pour l'extraction du message, ne peut pas toujours être garantie.

Asymétrique et Symétrique

Le marquage **asymétrique** est une technique qui utilise des paramètres différents pour l'insertion et l'extraction de la marque, et qui permet donc de la relire à l'aide d'une seule clef publique [WM01, KKC03].
Dans le marquage **symétrique** les paramètres utilisés pour insérer la marque sont les mêmes que pour l'extraire. Le rôle de clef-privée et clef-publique n'existe pas, l'insertion et l'extraction sont faites à l'aide de la même clef et procedure.

Fragile, Semi-fragile et Robuste

Dans le marquage **fragile**, la marque est fortement sensible aux modifications de l'image porteuse. Cette approche sert à prouver l'authenticité et l'intégrité d'un fichier tatoué. Nous citons les travaux [MC05, Li04, LXF01].
Le marquage **semi-fragile** a pour objectif de reconnaître les perturbations mal intentionnées et de rester robuste à certaines classes de dégradations légères de l'image, comme compression avec pertes par exemple. Diverses méthodes d'authentification d'images par marquage semi-fragile ont été proposées [LC98, RM02, ES04, MSCS06].

Le marquage **robuste** dispose d'un large champ de théories et de résultats. Celui-ci cherche à préserver les données cachées face aux attaques. La marque doit donc être suffisamment résistante aux attaques afin de rester identifiable [RK05, DFHS03, JA03, BC00].

Visible et Invisible

Le marquage **visible** est sujet à controverse. Il y a une branche de chercheurs qui disent que si la marque est visible, alors elle peut être facilement attaquée. Néanmoins, nous trouvons des applications qui demandent que la marque soit visible, c'est le cas du logo des sociétés dans les programmes télévisuels. Dans la catégorie de marquage visible, nous distinguons les travaux [MRRN04, HHKC03, MRK00].
Le marquage **invisible** est l'approche la plus développée qui attire la plupart des chercheurs. La majorité des techniques concernant la protection de propriété intellectuelle suit la branche du marquage invisible.

Inversible et Non-inversible

Le marquage **inversible** permet de récupérer toutes les propriétés originales de l'image porteuse après l'extraction de la marque [FD01, Li05, ZSW04].

Dans le marquage **non-inversible**, l'image originale est définitivement altérée par le mécanisme d'incrustation de la marque. La matrice originale de pixels est irrécupérable. La plupart des méthodes citées jusqu'ici sont non-inversibles.

Additif et Substitutif

Dans le marquage **additif**, le message à ajouter n'est pas corrélé à l'image couverture. La plupart des techniques de marquage aveugle est basée sur une insertion additive [GP03, BRM03, BB04].
Le marquage **substitutif** modifie les bits de la couverture afin de les faire correspondre à la marque. Ce type de marquage est connu comme marquage par contrainte, parce qu'il force l'image couverture à respecter certaines propriétés qui déterminent la marque [EBTG03, CMB02, BB04].

2.3.2 Modèles perceptifs pour le marquage d'images fixes

Un des grands soucis dans le marquage d'image est la détermination de la force d'insertion du marquage. Un marquage fort rend la marque plus robuste aux bruits et facilite le procédé de détection. Cependant, une insertion très puissante peut apporter des modifications visuelles importantes à l'image qui peuvent briser l'exigence d'imperceptibilité. Des règles, en prenant en compte le SVH, ont été proposées pour la répartition des distorsions à travers l'image. Il existe deux approches : la première est dans le domaine transformé et il s'agit de modifier certains coefficients en considérant le rapport entre la robustesse et l'invisibilité. La seconde approche, applicable aux schémas de marquage additif, se décompose en deux étapes. Dans un premier temps, le marquage (sur les caractéristiques d'un bruit blanc gaussien[1]) est conçu indépendamment de l'image et sans prendre en compte la contrainte d'invisibilité. Dans un deuxième temps cette marque est insérée de façon imperceptible dans l'image via un masque de pondération visuelle [JD03].

Utilisation de l'information de luminance

Ces modèles utilisent la propriété que notre oeil est peu sensible à des modifications de faible amplitude sur des régions de l'image où les variations de luminance sont importantes [DP03]. L'idée est de pondérer l'amplitude du marquage par une mesure d'activité locale de l'image. Cette mesure peut être accomplie de diverses façons : calcul de la variance locale, calcul de gradients spatiaux ou emploi du filtre laplacien.

D'autres modèles qui exploitent les caractéristiques des coefficients transformés se limitent aux modifications de certains coefficients qui peuvent rendre un masquage effectif. Pour la DCT par exemple, une modification des coefficients basses fréquences est très visible et une modification des coefficients hautes fréquences est très sensible aux attaques.

[1] Le bruit qui suit une loi normale de moyenne et variance données.

Utilisation de l'information de chrominance

Les modèles perceptifs qui utilisent des informations de chrominance tirent partie du fait que l'oeil humain est moins sensible aux modifications de chrominance que de luminance. Ce phénomène est aussi exploité par la norme JPEG où les composantes de chrominance sont sous-échantillonnées.

Certaines de ces méthodes se basent sur le fait que les changements d'espace couleurs enlèvent des dimensions de représentations qui sont moins perceptibles. La stratégie est donc de marquer directement sur le canal bleu parce que le SVH y est mois sensible. Il est également possible de prendre en compte les trois composantes couleurs avec différentes amplitudes de la marque par composante. L'amplitude de la marque dans la composante bleue α_B est environ 10 fois plus importante que dans la composante verte α_G, par exemple [DP03].

D'autres espaces de représentation des couleurs ont été utilisés comme les composantes IQ de l'espace YIQ et les composantes UV d'espace YUV [APFM01, PBJM03]. La plupart des techniques qui exploite l'information de chrominance pour les schémas de marquage insère aussi des informations dans la luminance. Ceci facilite la détection du message et rend le procédé robuste à une conversion en niveau de gris.

2.3.3 Manipulations et attaques sur les images

Il existe des manipulations appliquées sur les images qui peuvent apporter des distorsions importantes au message incrusté et changer son comportement. Ces manipulations peuvent être vues comme des attaques malveillantes (suppression, modification ou détérioration du message) ou simplement innocentes afin d'optimiser la qualité (filtres) ou l'enregistrement (compression) de l'image. Pendant longtemps les recherches dans le domaine du marquage ont été focalisées sur la protection des droits d'auteur et dont la robustesse est la propriété la plus importante. La marque doit être robuste à certains types de manipulations habituellement utilisées dans l'imagerie numérique. Nous allons présenter dans cette section une liste non exhaustive de ces manipulations.

Compression

La compression avec pertes est le mode de compression le plus utilisé. Elle a pour but de diminuer la taille du fichier image, comme détaillé chapitre 3. Les techniques de compression avec pertes suppriment les informations redondantes des images. Comme la marque n'est pas généralement visible, elle peut donc être considérée comme non significative et donc aussi être supprimée.

Rehaussement et lissage

Le rehaussement correspond à l'augmentation des composantes hautes fréquences de l'image. L'image devient alors plus contrastée. Le lissage est l'opération contraire du rehaussement, il atténue les composantes hautes fréquences de l'image qui devient alors plus floue. Ces opérations peuvent modifier également les composantes hautes fréquences du message et leur faire perdre leurs particularités.

Transformations géométriques usuelles

Parmi les transformations géométriques, la plus usuelle est la modification des dimensions de l'image et les transformations affines tels que la rotation, la translation et zoom. Ce genre de transformation provoque dans la plupart des cas une désynchronisation de la marque insérée lors de la détection et l'extraction.

Conversions analogique-numérique

La conversion analogique/numérique peut provoquer une perte de qualité ou ajouter du bruit dans l'image. Ceci peut être obtenu à partir de l'impression suivie d'une acquisition par scanner, ou encore à partir d'un film réalisé à l'aide d'un camescope dans une salle de cinéma. Ce type de conversion peut apporter de légères déformations géométriques sur les images provoquant une perte de synchronisation.

Modifications valumétriques

Il existe une catégorie de traitement (étalement d'histogramme, égalisation d'histogramme ou encore transformation Gamma) qui ne prend en compte que la luminance pour améliorer l'image. Comme le changement est fait sur la luminance, les informations marquées sur la chrominance peuvent être désynchronisées.

Débruitage

L'objectif de cette manipulation malveillante est d'approcher au mieux la forme d'onde du message, pour pouvoir l'enlever. Le message peut être estimé en utilisant le filtrage de Wiener. Cette estimation est alors soustraite à l'image originale pour l'obtention d'une copie du message.

Gigue

Le gigue (*Jittering*) est un phénomène connu en télécommunications. Lorsque le délai de transmission du signal varie, il en résulte une réplication ou une suppression d'un morceau du signal. Ceci peut se produire dans le domaine spatial ou temporel sur les images, il peut y avoir un ajout ou une suppression de lignes ou de colonnes.

Attaque par mosaïques

Il s'agit de découper l'image marquée en plusieurs morceaux. Cette attaque vise les moteurs de recherche automatique (*crawlers*) des marques dans les images sur Internet. L'image est ainsi envoyée par morceaux et assemblée dans une page HTML.

Stirmark

L'attaque Stirmark[2] est un logiciel qui permet d'apprécier la robustesse d'un procédé de marquage. Ce logiciel propose un banc de tests avec une grande variété de traitements sur les images, comme les manipulations présentées précédemment et plusieurs

[2] http ://www.petitcolas.net/fabien/watermarking/stirmark/

distorsions géométriques. La qualité de l'image résultante n'est pas dégradée, mais la marque est fortement modifiée ou effacée.

2.3.4 Techniques robustes aux distorsions synchrones

Les méthodes de marquage ont différents comportements selon la technique et le domaine d'insertion du message. La technique et le domaine de marquage peuvent rendre la marque plus ou moins résistante aux distorsions synchrones[3] ou asynchrones[4]. Une insertion par substitution apporte une robustesse plus importante mais ciblée sur un type de distorsion donnée comme par exemple une technique de compression. Dans cette section, les méthodes sont présentées en fonction de leur type et de leur domaine d'insertion.

Marquage additif dans le domaine spatial

Parmi les nombreux schémas spatiaux d'insertion basés sur l'ajout d'une séquence aléatoire 2D, nous distinguons la technique d'étalement de spectre de Hartung et Girod [FB98] et la technique du patchwork de Bender et Morimoto [BGM96]. La technique du ***patchwork*** consiste à diviser l'image en deux ensembles disjoints A_1 et A_2 de pixels qui dépendent d'une clef secrète. Ensuite les pixels de l'image, notés $p(i,j)$, sont modifiés différemment selon l'ensemble A_1 et A_2 auxquels ils appartiennent.

Marquage additif dans le domaine fréquentiel

Les méthodes classées dans ce groupe permettent d'obtenir une bonne robustesse à la compression JPEG. Les données sont insérées soit sur les blocs de pixels transformés soit directement sur la transformée de l'image complète.

Marquage additif dans le domaine multirésolution

Il existe de nombreuses méthodes de marquage qui agissent dans le domaine multirésolution [DD97, XCG97, WZY98].

Barni *et al.* ont proposé un schéma aveugle où le message est inséré dans les trois sous-bandes de détails (LH_0, HL_0, HH_0) pour avoir un meilleur résultat de la robustesse par rapport à l'invisibilité. Les coefficients des trois sous-bandes X^{LH_0}, X^{HL_0} et X^{HH_0} sont marqués par addition d'une séquence pseudo-aléatoire W de même taille que les trois sous-bandes.

Marquage substitutif dans le domaine spatial

Contrairement aux méthodes additives, qui dans l'ensemble ont des schémas similaires, les méthodes substitutives se distinguent par leur diversité. Dans le domaine spatial, l'insertion peut se faire par la modification des bits de poids faibles LSBs, par la quantification vectorielle spatiale, ou encore par l'insertion de similarités.

[3] Les manipulations qui ne modifient pas la position spatiale ou temporelle du signal.

[4] Les opérations qui changent la position spatiale ou temporelle du signal causant une désynchronisation.

La **modification du LSB** est une des techniques les plus simples. Le message est inséré dans l'image par le remplacement du LSB d'une composante couleur du pixel par le bit du message.

La **quantification vectorielle** consiste à remplacer des vecteurs de l'image (bloc de pixels) par des vecteurs appartenant à un dictionnaire prédéfini. Afin de restreindre l'impact sur l'image, les blocs du dictionnaire sont choisis de façon à être les plus proches possibles des blocs originaux.

Les méthodes basées sur l'**insertion de similarités** exploitent le changement des similarités existants entre les blocs de l'image afin d'insérer le message. Ainsi, des composantes de l'image sont substituées par des composantes qui possèdent une relation de similarité avec d'autres composantes [Bas00].

Marquage substitutif dans le domaine fréquentiel

La norme JPEG est composée de plusieurs étapes qui peuvent être utilisées pour le marquage d'information. L'insertion du message après le codage entropique (étape de compression) nous permet d'éviter d'effectuer une décompression totale de l'image lors de l'extraction.

La méthode **JPEG-JSTEG** [HW99, Upm02] insère le message sur les coefficients DCT quantifiés non nuls afin que la dégradation due à la compression soit négligeable. L'idée est d'utiliser le LSB de chaque coefficient quantifié dont la valeur est strictement supérieure à 1 pour insérer le bit du message. L'inconvénient de cette approche réside dans la forte dépendance de la capacité de stockage en fonction du contenu de l'image.

Marquage substitutif dans le domaine multirésolution

La transformation par ondelettes admet diverses perspectives pour l'insertion de données [CW01, GM03]. Les ondelettes permettent aussi de tatouer une image compressée au format JPEG2000 sans pour autant avoir à décompresser totalement l'image à tatouer.

2.3.5 Techniques robustes aux distorsions asynchrones

Une distorsion est dite asynchrone si elle détériore le synchronisme existant entre les données insérées et l'image couverture. Le repère d'insertion initial du message est déplacé et ceci pose des problèmes lors de la détection et de l'extraction. Quand une image marquée subit une distorsion asynchrone, elle doit alors être resynchronisée. Cette opération peut être extrêmement coûteuse en temps de calcul. Les techniques de distorsions asynchrones les plus couramment utilisées sont les transformations géométriques comme les rotations, translations et l'attaque *stirmark*, présentée section 2.3.3. Afin de faire face aux attaques asynchrones, plusieurs techniques ont été mises au point.

Insertion insensible à la géométrie

Afin de résister aux attaques géométriques l'approche générale consiste à insérer le message sans faire appel à la géométrie structurelle de l'image. Entre les nombreux

schémas insensibles à la géométrie, nous allons présenter les deux plus courant : la luminance et l'histogramme.

La **luminance** est une composante de certains espaces colorimétri1ques (comme YUV et CIELAB par exemple) qui possède des propriétés intéressantes pour le marquage. En effet, la luminance moyenne d'une image est conservée après des distorsions asynchrones [DP03]. Le schéma le plus utilisé est basé sur l'étalement de spectre. L'insertion du message est faite en ajoutant une composante continue générée à l'aide d'une séquence aléatoire. La détection est obtenue en calculant la corrélation entre la luminance moyenne et la séquence aléatoire utilisée lors de l'insertion.

L'**histogramme** représente le nombre d'occurences de l'image. Il est aussi, comme la luminance, peu sensible aux déformations asynchrones. L'insertion peut être faite par la reclassification des pixels en comparant leur valeur puis la moyenne des valeurs associées à des différents voisinages. Nous pouvons aussi rendre l'histogramme périodique pour insérer l'information.

Insertion périodique du message

L'insertion périodique d'information permet de réduire la complexité de la détection après une distorsion asynchrone. L'espace de recherche est alors réduit à la taille de la période de base. Un calcul de corrélation permet de récupérer le message après les distorsions asynchrones éventuelles. Toutefois, la redondance introduite rend ces méthodes facilement apparentes. De plus, elles ne sont pas robustes aux transformations géométriques non affines. **Delanay *et al.*** [DM00] ont suggéré une méthode qui génère un message périodique dont le support de base dépend d'une clef secrète.

Insertion d'un motif de resynchronisation

La stratégie ici est d'identifier directement la transformation géométrique afin de l'inverser et de détecter le message dans son repère initial. Ceci est obtenu par l'utilisation de motifs resynchronisants. **Kutter** [Kut99] propose une méthode dont l'information est insérée quatre fois dans l'image. Il devient ainsi redondant spatialement et donc repérable par auto-corrélation de l'image.

Utilisation de l'image originale

L'utilisation de l'image originale pour la détection (approche non-aveugle) aide l'identification de possibles transformations géométriques appliquées sur l'image tatouée. **Davoine *et al.*** [FPPJM99] ont proposé un schéma dont l'image originale et l'image tatouée sont manipulées pour compenser les déformations géométriques produites par stirmark. Dans leur approche, l'idée est de déplacer les différents sommets de la partition de l'image tatouée.

Utilisation du contenu de l'image

L'utilisation du contenu de l'image permet de construire un repère d'insertion spécifique qui subit les mêmes transformations asynchrones que l'image. Comme ce repère est lié au contenu de l'image il remporte une robustesse additionnelle.

Il existe plusieurs approches pour la description du contenu de l'image. Elle peut être faite par détection des contours ou de régions [AP05] ou encore par détection de points d'intérêts particuliers [BCM02]. **Lovarco *et al.*** [LVPD05] ont présenté une approche de descripteurs basée sur la détection de régions et sur le calcul du centre de gravité des objets de l'image. Les objets dans l'image sont identifiés et les coordonnées verticales et horizontales du barycentre de chaque élément sont calculées. Pour identifier la forme des objets, deux vecteurs propres indiquant les axes majeur et mineur sont calculés en utilisant une méthode dérivée de l'analyse en composantes principales. Ces vecteurs forment le repère pour l'insertion du message. Le message est inséré en blocs unitaires localisés autour de ces vecteurs plusieurs fois dans les différents espaces couleurs.

Conclusion

Dans ce chapitre, nous avons exposé les concepts et techniques d'insertion de données cachées. Nous avons pu remarquer que les approches existantes pour le marquage d'images sont nombreuses et variées. Chacune possède ses avantages et inconvénients. Le choix d'une méthode appropriée est fortement lié aux objectifs recherchés : robustesse, grande capacité (IDC), intégrité, authentification ou simplement transmettre un message (stéganographie). Une fois que l'objectif est défini nous pouvons alors utiliser l'association de différentes techniques. Dans les chapitres 4 et 5 nous présenterons des combinaisons de différentes approches de marquage et de compression d'images pour optimiser nos résultats.

La première méthode proposée dans le chapitre 4, se concentre sur les schémas de marquage : additif qui permettent d'assurer la réversibilité, fragile pour garantir l'intégrité et aveugle où l'image originale n'est pas requise au moment de l'extraction. Nous nous plaçons dans le cadre d'un schéma symétrique. L'état de l'art nous propose diverses alternatives de marquage basés sur des constatations empiriques, ou des schémas théoriques. Notre objectif est de proposer un schéma de marquage simple et facile à implémenter. Pour la deuxième méthode proposée au chapitre 4 nous employons un marquage également fragile, aveugle, mais substitutif pour tatouer une image cryptée.

Dans ce document, l'analyse de la qualité visuelle des images tatouées décryptées dans le chapitre 4 et des images traitées par cryptage sélectif (chapitre 5) est faite à l'aide du PSNR. D'autres méthodes pour mesurer la qualité visuelle (évaluation qualitative) des images qui prend en compte les phénomènes de perception humaine sont connues. Cependant, cette métrique est toujours d'actualité et il n'y a pas encore de standard d'évaluation qualitative.

Chapitre 3

Compression d'image

Sommaire :

3.1	Généralités sur la compression et l'images
3.2	Codage sans perte
3.3	Codage avec pertes
3.4	Domaine spatial
3.5	Domaine fréquentiel

Introduction

La qualité visuelle des images numériques augmente continuellement avec le développement de nouvelles techniques d'affichage (multirésolution, transmission progressive) et de nouvelles technologies d'acquisition (haute définition, nouveau hardware). Cependant, la taille de ces images augmente proportionnellement à leur qualité et leur stockage et transmission constituent donc les enjeux principaux dans le monde numérique. La compression s'impose comme une étape incontournable pour optimiser l'utilisation de ces grands volumes d'informations dans les réseaux informatiques. L'objectif principal de la compression d'image est de réduire la quantité d'information nécessaire à une représentation visuelle fidèle à l'image originale. Nous différençons les schémas de compression selon la perte d'informations. Les méthodes réversibles, section 3.2, utilisent uniquement le principe de la réduction de la redondance et n'engendrent pas de perte. Les méthodes irréversibles, section 3.3, définissent une représentation approximative de l'information.

Dans les sections suivantes, nous présentons quelques théories et aspects des nouvelles techniques de compression des images numériques.

3.1 Généralités sur la compression et l'images

Une image peut être représentée sous la forme vectorielle[1] ou sous la forme d'une matrice de points, *bitmap*. Dans ces travaux de thèse le terme image correspond au type bitmap ou matrice de points.

Une image est une matrice de $(M \times N)$ points appelés pixels et à chaque pixel est associé une ou plusieurs valeurs d'intensité qui se combinent pour déterminer la couleur.

3.1.1 Taux de compression et redondance

Le taux de compression soumis à une image est directement proportionnel à la quantité de redondance d'information qu'elle possède.

Taux de compression

Le taux de compression est utilisé pour mesurer le résultat d'un procédé de compression. Il est représenté soit comme une formule, équation (a), soit comme un facteur, équation (b). Dans les équations (a) et (b), I_o est la taille de l'image originale en octet et I_c la taille de l'image comprimée. Le taux de compression peut être aussi quantifié par le nombre moyen de bits par pixel (bpp), équation (c). L'élément $Bits_{I_c}$ est le nombre total de bits de l'image comprimée et $Pixels_{I_o}$ est le nombre total de pixels de l'image originale.

$$(\mathbf{a}) \quad \sigma = \frac{I_o}{I_c} \qquad (\mathbf{b}) \quad \sigma = (I_o/I_c) : (1) \qquad (\mathbf{c}) \quad \sigma = \frac{Bits_{I_c}}{Pixels_{I_o}}$$

[1] Les données sont représentées par des caractéristiques géométriques, comme traits, surfaces, etc.

Redondance

Une image numérique présente la particularité de posséder des corrélations importantes entre les pixels voisins. Cette corrélation est vue comme une redondance des informations pertinentes. La redondance peut être de deux natures : la **redondance spatiale** qui apparaît directement entre les pixels voisins de l'image originale et la **redondance spectrale** qui est liée aux fréquences et qui est acquise avec les transformations de domaines. La redondance dans le domaine spatial n'est pas facilement identifiable et généralement ne fournit pas toujours un bon taux de compression. Il est donc nécessaire de faire une transformation pour obtenir une décorrélation de l'information spatiale et un groupement d'énergie fréquentielle.

3.1.2 Critères psychovisuels et compression

Les méthodes de compression sans perte ne causent aucun problème visuel car ils sont totalement réversibles. Par contre, les procédures de compression avec pertes diminuent la qualité de l'image. Plus le taux de compression est important, plus les distorsions apparaissent dans l'image. Le point critique est la définition de la quantité de distorsions par rapport à la qualité de l'image. Le SVH (*Système Visuel Humain*) possède des caractéristiques particulières qui doivent être prise en compte. Notre oeil est capable de distinguer environ sept millions de couleurs. Quand nous regardons une image notre système visuel doit résoudre beaucoup de contraintes : perception 3D, ombres, objets cachés, etc. En fait, le SVH tente de donner un sens visuel à chaque objet. Notre perception est influencée par ce que nous nous attendons à voir, c'est le cas des illusions optiques par exemple.

La perception de la couleur est influencée fortement par la saturation et la luminance. La saturation est la quantité de blanc ajoutée à une couleur, et la luminance est la mesure de lumière réfléchie par un objet. L'oeil humain possède des sensibilités différentes suivant l'orientation des contrastes[2]. Il est beaucoup plus sensible à la luminance qu'à la chrominance. Les systèmes de compression avec pertes changent généralement d'espace couleur. Il est conseillé de ne pas dégrader beaucoup la composante de luminance. La norme JPEG, décrite section 3.5, par exemple, sous-échantillonne les deux composantes de chrominance, et ne change pas la composante de luminance.

Contours et Texture

La perception des **contours** fait partie des fonctions essentielles du SVH. Nous délimitons mentalement les objets qui sont dans une image grâce à leurs contours, et nous sommes très sensibles à la dégradation de ces contours. La théorie des formes explique leur importance pour la perception correcte des objets partiellement cachés. La fermeture des contours pour produire une forme visuelle connue par notre mémoire est essentielle.

[2]Il a la propriété de distinguer deux régions distinctes dans une image, à l'aide de la différence de couleur et de luminosité.

3.2 Codage sans perte

La compression sans perte ou codage entropique ou codage réversible permet de retrouver la valeur exacte du signal comprimé lorsqu'il n'y aucune perte de données sur l'information d'origine. En fait, la même information est réécrite d'une manière plus concise. Le processus de codage sans perte crée des "mots-codes" à partir d'un dictionaire statique ou d'un dictionaire construit dynamiquement. Ces processus s'appuient sur des informations statistiques de l'image. Les codes statistiques les plus répandus sont le codage d'Huffman et le codage arithmétique. Le **codage statistique permet** de s'approcher au mieux de l'entropie [RCG02]. Ils ont pour principe d'associer aux valeurs les plus probables les mots binaires les plus courts.

3.2.1 Codage d'Huffman

Huffman [Huf52] a suggéré une méthode statistique qui permet d'attribuer un mot-code binaire aux différents symboles (pixel) à compresser. La probabilité d'occurrence du symbole dans l'image est prise en compte en attribuant aux plus fréquents des codes courts, et aux plus rares des codes longs, VLC - *Variable Length Coding*. La suite finale de pixels codés à longueurs variables sera plus petite que la taille originale. Le codeur Huffman crée un arbre ordonné à partir de tous les symboles et de leur fréquence d'apparition. Les branches sont construites récursivement en partant des symboles les plus fréquents. Le code de chaque symbole correspond à la suite des codes le long du chemin allant de ce caractère à la racine. Plus le symbole est profond dans l'arbre plus la quantité de bits pour le représenter est important. La figure 3.1 présente un exemple de codeur d'Huffman. Le tableau de gauche montre que nous avons au total 21 symboles qui sont représentés en octets équivalant à 168 bits. Après la construction de l'arbre d'Huffman nous pouvons constater un taux de compression de $\sigma = \frac{45}{168}$ de $26,79\%$.

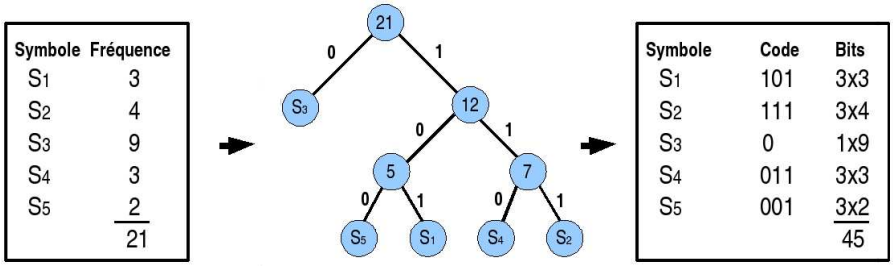


FIG. 3.1 – Algorithme d'Huffman.

3.2.2 Codage arithmétique

Le codage arithmétique (CA) [HV94] est un codage statistique qui attribue à une suite de symboles une valeur réelle. Il consiste à découper l'intervalle des réels [0, 1) en sous-intervales, dont les longueurs sont fonctions des probabilités des symboles. Le codage arithmétique n'attribue pas un code à chaque symbole comme Huffman et les

autres codages par blocs, mais un code au message tout entier. Les tableaux suivants présentent un exemple de codage arithmétique avec le message **AAOEU**.

Alphabet	Probabilité	Probabilité Cumulée	Partition initiale
A	0,2	0,2	[0 0,2)
E	0,4	0,6	[0,2 0,6)
I	0,1	0,7	[0,6 0,7)
O	0,2	0,9	[0,7 0,9)
U	0,1	1,0	[0,9 1,0)

Message	Gauche G	Taille T	Droite D
A	0,0000	0,2000	0,2000
A	0,0000	0,0400	0,0400
O	0,0280	0,0080	0,0360
E	0,0296	0,0032	0,0328
U	0,0325	0,0003	0,0328

TAB. 3.1 – Codage arithmétique

Soit l'alphabet {A,E,I,O,U} avec les probabilités {0,2 0,4 0,1 0,2 0,1}. Le codage arithmétique est fait à partir de l'intervalle initial [0, 1) et au fur et à mesure du codage, la longueur de l'intervalle diminue en tenant compte du sous-intervalle précédent. Nous nous servons des formules ($G = G_I^P + G_M * T_I^P$) et ($T = T_I^P * T_M$) pour construire le nouvel intervalle où les lettres signifient : G-gauche, T-taille, M-message et P-précédant. Le premier symbole **A** du message réduit l'intervalle initial à [0 0,2). Le deuxième symbole **A** du message réduit ce dernier intervalle à [0 0,04) (1/5 de l'intervalle précèdent). Le symbole **O** réduit l'intervalle à [0,028 0,036). Le symbole **E** diminue l'intervalle à [0,0296 0,0328). Enfin, le symbole final **U** réduit à [0,03248 0,0328). Finalement, tout réel dans l'intervalle [0,03248 0,0328) codera le message AAOEU. Le codage arithmétique est présent dans la norme JPEG (dans les modes *Extended DCT-based processes* et *Lossless processes*) et JPEG2000.

Les méthodes de codage statistiques construisent les mots-codes à partir d'un dictionnaire prédéfini, basé sur les statistiques de l'image elle-même. Ce dictionnaire est indispensable pour le décodage. Des nouvelles études et améliorations pour cette approche ont été proposées, nous citons les travaux [FHF02, GFG01, SLL01].
Les deux méthodes que nous allons présenter, **codage par substitution**, n'exigent pas de connaissance à *priori* de l'image, comme les probabilités d'apparition des pixels par exemple. Elles construisent des dictionnaires dynamiques dont les mots-codes créés sont indépendants de la source.

3.2.3 L'algorithme LZW

Lempel et Ziv [ZL77] ont présenté un schéma (LZ77) qui est à la base de tous les algorithmes à dictionnaire dynamique utilisés actuellement. Welch a amélioré leur algorithme et a déposé un brevet en créant l'algorithme LZW qui génère un dictionnaire dynamique qui contient des motifs du fichier. L'utilisation d'un dictionnaire dynamique a réglé le problème de le transmettre *a priori* pour les procédés de compression et décompression. Il est basé sur la multiplicité des occurrences de séquences de symboles. Son principe consiste à substituer des motifs par un code d'affectation en construisant au fur et à mesure un dictionnaire. Celui-ci est initialisé avec les valeurs de la table

ASCII. Chaque octet du fichier est comparé au dictionnaire. S'il n'existe pas, il est ajouté au dictionnaire. L'algorithme LZW fait partie du format d'image, aussi breveté, GIF - (*Graphics Interchange Format*).

3.2.4 Codage par plage

Le codage par plage ou RLE *Run Length Encoding* est recommandé lorsque nous observons des répétitions de symboles consécutifs. Il est utilisé par de nombreux formats d'images (BMP, TIFF, JPEG) [RCG02]. L'idée est de regrouper les pixels voisins ayant la même couleur. Chaque groupement définit un couple de valeurs $P = (plage, n)$ où *plage* est le nombre de points voisins ayant la même valeur, et n est cette valeur. Le RLE est d'autant plus performant que les groupements sont étendus, il n'est pas applicable dans tous les cas. Il est recommandé pour les images avec de larges zones uniformes. La compression d'une image peut être effectuée de manière adaptative : dans les régions uniformes le RLE est appliqué, et dans les zones non uniformes des règles particulières sont créées. Par exemple, au moins trois éléments se répètent consécutivement alors la méthode RLE est utilisée, sinon un caractère de contrôle est inséré, suivi du nombre d'éléments de la chaîne non compressée. D'autres caractères de contrôle peuvent aussi être utilisés pour définir la fin de ligne ou la fin de colonne.

3.2.5 Codage par prédiction linéaire

Les algorithmes qui utilisent le codage par prédiction exploitent la redondance spatiale. Il s'agit de prédire la valeur d'un pixel en fonction de la valeur des pixels voisins et de ne coder que l'erreur de prédiction. Le gain en compression est accompli par la variation faible entre pixels voisins, sauf pour les pixels situés sur les contours. Le voisinage peut être défini selon sa connexité (4-connexité ou 8-connexité) ou selon l'ordre du parcours choisi pour accéder aux pixels voisins. L'une des techniques de prédiction la plus simple est la DPCM (*Differential Pulse Code Modulation*) [Mor95]. Cette technique effectue une prédiction à base d'une combinaison linéaire des valeurs des pixels voisins. Une version adaptative, ADPCM , qui utilise différentes formes de prédiction et de voisinage selon le contexte et le contenu de l'image a été présentée par Kyung *et al.* [JGB96]. Récemment, Babel *et al.* [BDR03] ont proposé un codage progressif et multirésolution LAR - (*Locally Adaptive Resolution*). Il s'agit d'un codeur qui associe le DPCM à une décomposition multi-couches suivi d'une transformée Mojette[3]. La profondeur de la décomposition détermine le type de compression, avec pertes ou sans perte pour le huitième niveau.

[3]C'est une transformation qui projette des informations disposées en 2D sur des vecteurs 1D.

3.3 Codage avec pertes

Les méthodes avec pertes (*lossy*) ou irréversibles sont des méthodes qui tirent parti d'une corrélation (ou redondance) existante dans l'image. L'information perdue est due à l'élimination de cette redondance, ceci rend possible une compression plus importante. La perte d'information est toujours discutable et nous nous posons alors la question de la limite acceptable. Cette limite est définie par le type d'application, comme les images médicales ou satellites par exemple. La quantification est un des mécanismes utilisé dans les algorithmes de compression, qui produit des pertes d'information.

3.3.1 Quantification

La quantification fait partie de plusieurs méthodes de compression d'image. L'objectif est de réduire la taille des coefficients de façon que cette réduction n'apporte pas de dégradations visuelles à l'image.

Scalaire

La quantification scalaire SQ - (*Scalar Quantization*) est une procédure qui associe à une variable continue X une variable discrète x. Pour cela on associe à x la valeur quantifiée $x_q = Q(X)$, où Q est une fonction (non linéaire) de quantification de $\mathbb{R} \rightarrow \mathbb{Z}$ [BS04]. La quantification scalaire utilisée, en pratique, dans les images sont des quantifications basées en *zone morte* dans lesquelles l'intervalle de quantification est centré à l'origine et est de taille multiple de la taille des autres intervalles de quantification [TM01].

Vectorielle

La quantification vectorielle VQ - (*Vector Quantization*) a été développé par Gersho et Gray [GG92] et elle fait aujourd'hui l'objet de nombreuses publications dans le domaine de la compression numérique [SZL95]. Le principe de la quantification vectorielle est issu du travail de Shanon qui montre qu'il était toujours possible d'améliorer la compression de données en codant non pas des scalaires, mais des vecteurs. Un quantificateur vectoriel Q associe à chaque vecteur d'entrée $X_i = (x_j, j = 1 \ldots k)$ un vecteur $Y_i = (y_j, j = 1 \ldots k) = Q(X_i)$, ce vecteur Y_i étant choisi parmi un dictionnaire (*codebook*) de taille finie. La VQ produit de meilleurs résultats que la SQ, néanmoins la VQ nécessite un codage complexe et de grandes capacités de mémoire.

3.3.2 Codage prédictif avec pertes

Il existe des techniques qui exploitent la redondance spatiale, cependant la prédiction est faite par approximation. Ces algorithmes ont comme objectif de rechercher un modèle de représentation le plus adéquat de l'information à coder afin d'obtenir un coût de codage minimal. L'idée est de coder l'erreur de prédiction au-dessus d'un seuil. Ce seuil

peut être défini par rapport à la qualité de l'image ou le niveau de compression espéré. Nous distinguons les travaux [VKG95, Pat98].

3.3.3 Codage par transformation

Les méthodes qui utilisent cette technique utilisent des transformations pour produire une décorrélation des redondances spectrales. Les pixels passent d'un espace où ils sont fortement corrélés dans un autre espace où leur corrélation est moindre. Lors de chaque transformation, le signal d'origine est remplacé par sa représentation dans un autre domaine. Dans divers algorithmes cette transformation d'espace est accompagnée d'une quantification et d'un codage entropique pour accomplir la compression de l'image. Ceci est le cas des normes standards de compression : l'algorithme JPEG, section 3.5.1.1 qui utilise la transformation type DCT et l'algorithme JPEG2000, section 3.5.1.3 qui utilise la transformation en ondelettes DWT.

3.4 Domaine spatial

Les méthodes de compression d'images dans le domaine spatial exploitent la redondance entre un pixel et son voisinage, ou entre certaines régions de l'image.

3.4.1 Quadtree

La technique de décomposition par quadtree est basée sur une approche récursive d'un codage arborescent. Une image hétérogène de taille $2^n \times 2^n$ est alors divisée en quatre sous-régions, de taille $2^{n-1} \times 2^{n-1}$. Ce processus est refait jusqu'à ce que chacune des régions soit déclarée homogène selon un critère choisi. Certains travaux proposent d'optimiser ce compromis en découpant l'image en blocs de tailles variables [UIO02, CLC+01]. Des zones considérées homogènes seront découpées en grands blocs alors que des zones texturées ou contenant des contours seront découpées en blocs plus petits comme le montre la figure 3.2.

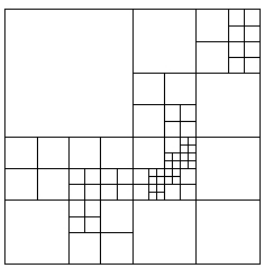


FIG. 3.2 – Décomposition en quadtree.

3.4.2 Décomposition en plans binaire

Cette technique est principalement employée pour les images en niveau de gris. L'idée est de décomposer l'image en huit images binaires, une pour chaque bit de niveau de gris, en commençant par les bits de poids les plus forts, voir figure 3.3.

FIG. 3.3 – Décomposition en plans binaires des bits de poids forts jusqu'aux bits de poids faibles.

Sur chaque image binaire sont lancées des procédures particulières pour effectuer la compression. L'image binaire représentant le bit de poids le plus fort, première image (8), apporte le plus haut taux de compression, tandis que la compression sur l'image binaire de poids le plus faible, dernière image (1) a quasiment une compression nulle.

Maniccamam et Bourbakis [MB01] ont présenté une méthode de compression sans perte basée sur la décomposition de l'image en plans binaires et sur la méthodologie SCAN[4]. Dans leur approche, chaque plan binaire est également décomposé en régions, et chaque région est parcourue selon un ensemble de chemins différents. Cet ensemble de chemins est composé de 32 chemins standards de plus d'autres chemins sont générés selon le langage SCAN. Pour chaque chemin de cet ensemble est réalisé un codage RLE pour déterminer le taux de compression optimal. La compression de l'image est déterminée pour l'ensemble des compressions de chaque région.

Dans le chapitre 4, nous présentons une nouvelle méthode de crypto-compression basée sur la décomposition en plans binaires et compression RLE.

3.4.3 Fractales

La compression par fractale est une technique de compression avec pertes encore peu utilisée. Une fractale est une structure géométrique qui se reproduit, dans une boucle infinie, par transformation affine (translation, rotation et mise à l'échelle). Cette structure se refait à toutes les échelles de forme réduite et légèrement déformée. La compression par fractale est basée sur le principe qu'il existe des similarités entre différentes régions isolées d'image. Elle exploite les récurrences des motifs qui, après quelques traitements, peuvent permettre une compression. La figure 3.4 présente un exemple d'exploitation des motifs. Dans la compression par fractales nous distinguons les travaux [IdSC06, DP03].

[4]Le SCAN est un langage formel basé sur l'accès spatial en deux dimensions.

FIG. 3.4 – Compression par fractales.

3.5 Domaine fréquentiel

Les techniques de compression dans le domaine fréquentiel s'appuient sur une transformation de l'image vers un nouvel espace de représentation d'énergie fortement décorrélée. Cette décorrélation provoque une nouvelle représentation de l'image par la redistribution de l'énergie dans un nombre restreint de coefficients transformés. Cette énergie de l'image transformée est distribuée sous la forme de tranches énergétiques de basse, moyenne et haute intensités. Les transformations d'espace les plus courantes sont la DCT détaillés dans l'annexe A et la DWT.

3.5.1 Les standards

Il existe plus d'une cinquantaine de types de formats d'image [RCG02]. Pour chacun d'entre eux la structuration des données et les attributs sont différents. La standardisation d'un format d'image permet de régler l'utilisation, la divulgation et la production de logiciels et de hardware compatibles avec le format standard. Le format standard JPEG est le format d'image le plus populaire, et il est devant la scène depuis quelques années. Son successeur, le JPEG2000, semble s'établir dans le domaine de l'image numérique. Le JPEG2000 possède des fonctionnalités supplémentaires par rapport au format JPEG. Cependant, la plupart des appareils numériques (appareils photos, camescopes, téléphones portable, etc) et les logiciels qui capturent et traitent les images sont au format JPEG.

3.5.1.1 JPEG

Le comité ***Joint Photographic Expert Group*** a été créé en 1986 par la jonction (*Joint*) de plusieurs groupes qui travaillaient sur le photographie. Ce comité a produit la norme de compression d'images photographiques qui a été standardisée (ISO/IEC/10918-1/ 1994) et a reçue son nom JPEG. Il est devenu le format le plus populaire très rapidement parce qu'il a été conçu avec différentes contraintes :

- L'algorithme JPEG doit être implémentable sur une grande variété de types de CPU (unité centrale de calcul) et sur des cartes plus spécialisées (appareil photo numérique et téléphone portable par exemple).

– Il doit pouvoir compresser efficacement tout type d'images réelles (images photographiques, médicales) avec pertes et sans perte.
– Il possède quatre modes de fonctionnement : séquentiel (*baseline*), progressif (*extended DCT-based*), sans perte *Lossless*, hiérarchique *hierarchical*.

Entre les 4 modes de compression de la norme JPEG, le séquentiel ou *baseline* est le mode principal le plus répandu. Il est basé sur la transformation DCT, quantification scalaire et le codage d'Huffman sur pixels de 8 bits par plan de couleur.

Dans le chapitre 5 nous allons présenter un nouveau schéma de cryptage sélectif basé sur le mode baseline. Il nous semble donc important de décrire en détail le fonctionnement de ce mode. La figure 3.5 expose une synthèse du mode séquentiel.

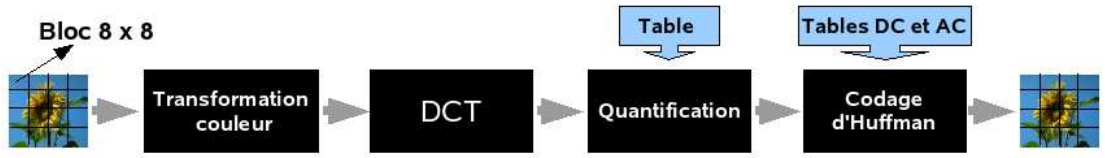


FIG. 3.5 – Compression JPEG.

– **Transformation d'espace couleur**
Tout d'abord, l'image originale est soumise à un changement d'espace couleur YCbCr[5]. Les informations de chrominance, les plans Cb et Cr, sont sous-échantillonnées. Après cette intervention, le deux plans auront deux fois moins de lignes et de colonnes. Cette opération de sous-échantillonage est fondée sur le principe que le SVH ne peut discerner des différences de chrominance au sein d'un carré de 2×2 points. Après le sous-échantillonage, chaque plan Y, Cb et Cr est traité de la même manière selon le schéma fonctionnel présenté figure 3.5. Tout d'abord, ils sont découpés en blocs de 8×8 pixels.

– **DCT** (Annexe A)
Chaque bloc 8×8 est soumis à une transformation par DCT. Le premier coefficient de la DCT, le DC, est proportionnel à la moyenne des valeurs du bloc. Les 63 autres coefficients sont appelés AC. Ce nouveau domaine transformé permet une décorrélation très forte de l'information. Sur ce bloc de coefficients, les énergies sont groupées en basse (zone homogène de l'image originale), moyenne et haute fréquences (zone texturée ou zone de contour). Avec la DCT, chaque colonne est une fonction cosinus de fréquence différente. La variance est alors concentrée sur les composantes de basse fréquence, et les composantes de haute fréquence seront annulées par quantification.
– **Quantification**
La compression avec pertes est faite dans l'étape de **quantification** qui est réalisée à l'aide d'une matrice Q de quantification de 8×8 éléments, figure 3.6.b. L'image subit des distorsions selon le niveau de compression désiré. Chaque coeffi-

[5]Le Y est la luminance, Cb et Cr sont les chrominances.

cient DCT est divisé par la valeur correspondante dans Q et le résultat est arrondi à l'entier le plus proche. L'acuité du SVH est plus faible à des hautes fréquences et plus sensible aux basses fréquences. Quelques tables standards, pour la quantification, ont été générées grâce à une série de caractéristiques psychovisuelles. Les valeurs prennent donc en compte cette caractéristique et introduit majoritairement de la distorsion dans les hautes fréquences. Cependant, il est tout à fait possible de personnaliser la table [Mon96].

– **Codage d'Huffman**
Le codage entropique dans le mode séquentiel est un codage du type RLE. Après la quantification un grand nombre de coefficients sont nuls ou très proches de zéro. Le coefficient DC (composante continue) est codé séparément par rapport aux AC. Les coefficients AC sont parcourus en zigzag, figure 3.6.a, et sont codés par des couples (HEAD),(AMPLITUDE). L'entête HEAD contient des contrôleurs qui seront utilisés pour accéder aux tables d'Huffman. Le paramètre AMPLITUDE est un entier signé correspondant à l'amplitude du coefficient AC non nul. La structure HEAD varie en fonction du type de coefficient. Pour les AC elle est composée de (RUNLENGTH, SIZE), alors que pour les DC elle est composée seulement de la taille SIZE. Les coefficients DC transportent une information visible importante et une corrélation locale significative. Ils sont hautement prédictibles, ainsi JPEG traite les coefficients DC séparément des 63 coefficients AC. La valeur des composants DC est importante et variée, mais est souvent très proche de celle de ses voisins. La seule valeur qui est donc encodée est la différence DIFF entre le coefficient DC_i quantifié du bloc courant et le précédent DC_{i-1}.

La figure 3.6 exhibe un bloc 8×8 de coefficients DCT quantifiés et le chemin en zigzag utilisé par le codage d'Huffman.

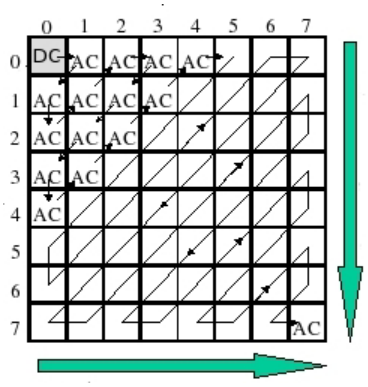


FIG. 3.6 – Bloc DCT et parcours zigzag.

3.5.1.2 JPEG-LS - [ISO/IEC/14495-1/ 1999]

La compression JPEG en mode sans perte n'est pas optimisé. Les objectifs du comité étaient de concevoir un mode réversible permettant une compression de l'image à 50%. Malheureusement, un code réversible est quasiment impossible avec l'utilisation de la DCT. Les erreurs d'arrondis dûes à la précision limitée de calcul sont toujours présentes. Le nouveau standard de compression sans perte, le JPEG-LS est basé sur une variation de la méthode LOCO-I - (*low complexity lossless compression method*) [WSS00]. Dans JPEG-LS le procédé de compression est composé de trois parties : la **prédiction** de la valeur du pixel qui est faite par rapport aux pixels voisins en utilisant l'approche de prédiction MED (*Median Edge Detection*) ; la **détermination d'un contexte**. Ce contexte représente l'environnement du pixel à coder et ses voisins. L'idée est de prendre le meilleur environnement qui affine la prédiction avant le codage et de réduire le nombre de paramètres de l'erreur de prédiction ; le **codage de l'erreur de prédiction** dont l'approche est de réinsérer l'erreur de prédiction dans le système puis de la comparer à d'autres mesures d'erreur.
Le JPEG-LS possède une option de haut taux de compression, mais cette option est quasiment sans perte (*near lossless*). Les observations expérimentales montrent que pour des taux supérieurs à 1,5 bpp, le JPEG-LS en mode *near lossless* donne de meilleures performances que JPEG.

3.5.1.3 JPEG2000 - [ISO/IEC/15444-1/ 2000]

Le JPEG2000 [TM01] remplace le JPEG comme le format standard pour la compression des images. Il a été réalisé dans la perspective de répondre aux exigences des nouvelles applications les plus diversifiées, comme la multirésolution par exemple.
La compression JPEG2000 est composée de plusieurs étapes selon les schémas avec pertes et sans perte. Tout d'abord, un **changement d'échelle** est effectué dans chaque composante couleur RGB, l'échelle est changée de (0, 255) à l'échelle (-128, 127) par une simple soustraction de 128 de chaque valeur. Après le changement d'échelle, l'image est soumise à une **transformation de plans couleurs**, facultative, de RGB à YCbCr. Cette transformation peut être réversible (sans perte) ou irréversible (avec pertes). Chaque plan chromatique de l'image est découpé en petites images appelées **tuiles**, *tile*. Chaque tuile est considérée comme une image et est traitée de façon indépendante. Du fait de la complexité du mécanisme du JPEG2000, la décomposition de l'image en tuiles rend possible l'application du JPEG2000 sur des images de taille importante.
La **transformation de domaine** est faite à l'aide d'une décomposition en ondelettes par schéma *lifting* [Swe96]. Chaque tuile de chaque composante subit des transformations selon les types compressions réversible et irréversible. Pour la compression réversible la norme utilise la paire (5,3) de LeGallles. Les coefficients des filtres d'analyse sont entiers et composés de 5 coefficients pour le filtre passe-bas et de 3 pour le passe-haut. Pour la compression irréversible est utilisée la paire (9,7) de Daubechies qui utilise des coefficients réels, avec 9 coefficients pour le passe-bas et 7 pour le passe-haut. Les lignes et les colonnes sont récursivement décomposées générant quatre sous-bandes pour chaque niveaux de décomposition. Ceci produit des sous-bandes LL (basses fréquences

horizontales et verticales), LH (basses fréquences horizontales et hautes fréquences verticales), HL (hautes fréquences horizontales et basses fréquences verticales) et HH (hautes fréquences horizontales et verticales).
Ensuite, une étape de **quantification** est faite selon le taux de compression. La quantification est faite pour rendre nuls les coefficients les plus faibles et faciliter le codage des autres. Chacune des sous-bandes peut être quantifiée avec des étapes différentes. Plusieurs techniques permettant la quantification des sous-bandes ont été proposées, linéaire, non-linéaire, avec masquage en fonction des voisins ou codage en treillis (TCQ), mais seule la quantification linéaire a été retenue par la norme.
Avant le **codage entropique**, les coefficients quantifiés sont divisés en plusieurs blocs appelés *code-blocks* de taille 64x64 ou 32x32. Le codage entropique du JPEG2000 est du type arithmétique adaptatif avec contexte, EBCOT - (*Embedded Block Coding with Optimized Truncation*) [TM01]. Le codage d'un bloc consiste à parcourir les coefficients du code-block par plan de bits, du bit de poids le plus fort (MSB) au bit de poids le plus faible (LSB). Les bits sont séparés en trois groupes en fonction de leur voisinage. Ensuite, ils sont codés en trois passes (*coding passes*). Le codage débute lorsqu'un plan de bits devient significatif (*signifiant*).
La résistance aux erreurs est une caractéristique particulière du JPEG2000. Après le codage entropique plusieurs caractères de contrôle (*segment marks, resynchronising marks*) sont insérés dans le flux de bits. Cette démarche est faite pour synchroniser les informations, limiter la taille du segment et éviter la propagation des erreurs.
Une autre fonctionnalité importante du JPEG2000 est la compression par région d'intérêt (ROI). Ceci permet d'avoir des taux de compression différents dans certaines régions de l'image. Les zones importantes peuvent être compressées quasi sans pertes et les zones moins importantes avec un fort taux de compression.
Malgré ces nombreuses fonctionnalités, le JPEG2000 possède quelques inconvénients. Il nécessite entre deux et six fois plus de cycles de CPU que JPEG[6] et il n'est pas indiqué pour les machines avec faibles ressources comme les appareils photos numériques par exemple. L'algorithme JPEG est beaucoup moins complexe et il peut être implémenté en hardware.

Conclusion

Dans ce chapitre, nous avons présenté plusieurs techniques de compression d'images sans perte et avec pertes. Tout d'abord, nous avons donné quelque concepts et mesures pour la compression d'images. Nous avons vu qu'un système de compression d'images se décompose fondamentalement en : acquisition, pré-traitement, quantification et codage entropique. D'une manière générale, toutes les approches énumérées précédemment utilisent d'une manière ou d'une autre les corrélations entre pixels voisins dans l'image. Ces corrélations sont le fondement de la compression car elles sont liées à la notion de redondance. Nous avons vu qu'après un changement de domaine les énergies sont groupées et que la transformation est généralement accompagnée d'une quantification. La quan-

[6] http ://sic.epfl.ch/publications/FI01/fi-3-1/3-1-page1.html.

tification rend les composantes de hautes fréquences nulles et celles-ci présentent alors des plages de valeurs nulles qui suggèrent l'utilisation de codage à longueur variable. Enfin, la connaissance des caractéristiques du SVH est très importante pour conserver un rapport qualité/compression correct et pour augmenter la compression par une pondération psychovisuelle.

Dans les standards actuels peu de fonctionnalités proposent de combiner compression et cryptage ou compression et marquage. Dans la seconde partie de ce document nous monterons comment développer des systèmes hybrides.

Nous utilisons du codage par plage pour accomplir de la compression dans nos méthodes proposées au chapitre 4. Le codage par plage c'est un codage simple, rapide et utilisé par de nombreux formats d'images comme le BMP, TIFF et JPEG par exemple.

Au chapitre 5 nous utilisons le format standard JPEG. Notre choix a été basé sur la popularité de ce format. Le JPEG est encore l'algorithme le plus utilisé pour la compression d'images. Il a été développé sur des quantités de cartes dédiées à la compression pour de nombreux dispositifs numériques. .

Deuxième partie

Transfert sécurisée d'images par combinaison de techniques de cryptage, marquage et compression

Seconde partie

Transfert Sécurisée d'images par combinaison de techniques de cryptage, marquage et compression

L'objectif de cette partie est de présenter les trois méthodes de protection d'images développées au cours de ma thèse. Les deux premières méthodes combinent les technologies de cryptage, marquage et compression de données pour le transfert sécurisé. Nous présentons également une nouvelle approche de cryptage sélectif pour la dissimulation/protection réglable (*scalable*) des informations dans les images numériques.

Introduction seconde partie

Les trois domaines de recherche abordés dans la première partie (cryptage, insertion de données cachées et compression des images numériques) servent de support théorique et de mécanisme d'évaluation pour la seconde partie de cette thèse.

Cette seconde partie est composée de deux chapitres, où nous allons présenter trois méthodes, qui associent les technologies mises au point au cours de la thèse. Le chapitre 4 présente deux méthodes. La première méthode proposée permet de combiner un chiffrement basé sur le mélange (*scrambling*), une compression par codage par plage et l'insertion de données cachées additif en créant un nouveau format d'image. Cette méthode de crypto-compression avec IDC apporte aucune perte d'information à l'image originale ni au message. Nous montrons dans cette méthode qu'en découpant l'image en deux parties (deux images semi-pixels) il est possible dans la partie haute (plans binaires de poids forts) de l'image d'effectuer à la fois de l'insertion de données cachées et de la compression.

La seconde méthode du chapitre 4 combine cryptage d'images et insertion de données cachées afin de rendre autonome un système de transmission sécurisée d'images. Nous avons rappelé que les méthodes asymétriques ne conviennent pas au chiffrement des images car elles sont trop longues en temps de calcul. Nous devons donc utiliser un chiffrement classique à clef secrète. Cependant, il faut utiliser un autre canal de transmission pour transférer la clef. Nous avons donc développé un algorithme de chiffrement par flot asynchrone robuste au bruit. Nous proposons d'insérer (par marquage substitutif) dans l'image cryptée la clef secrète chiffrée par l'algorithme RSA, présenté section 1.2.1.

Le chapitre 5 expose une nouvelle approche de cryptage sélectif basée sur les travaux de Droogenbroeck et Benedett [DB02] pour les images au format JPEG. Le cryptage sélectif permet de crypter de manière sélective les données d'une image tout en conservant un niveau de sécurité suffisant. Le cryptage de notre approche est également basé sur le changement du vecteur d'Huffman dans le codage entropique du JPEG. Néanmoins, notre approche rend possible le cryptage et le décryptage réglable de l'image ou d'une région d'intérêt. Nous montrons également l'application de notre approche dans plusieurs domaines comme les images médicales, peintures numériques et la protection de visages dans des séquences d'images.

À la fin de chaque chapitre nous faisons un bilan des avantages et des inconvénients des méthodes développées.

Chapitre 4

Codage Hybride

Sommaire :

4.1	Méthode réversible de crypto-tatouage appliquée aux images médicales
4.2	Méthode autonome de masquage de données

4.1 Méthode réversible de crypto-tatouage appliquée aux images médicales

Introduction

La plupart des méthodes d'IDC sont des méthodes dites irréversibles ou avec pertes. Cela signifie qu'après l'IDC, l'image porteuse a été modifiée. Nous ne pouvons donc pas retrouver l'image originale et certaines informations importantes peuvent être perdues. Pour des applications particulières telles que l'imagerie médicale, l'image originale doit absolument être conservée. Les méthodes d'IDC réversibles sont la solution pour ce type de problème. Ces méthodes peuvent être utilisées pour associer à l'image des informations cruciales sans changer le contenu de l'image. Plusieurs méthodes ont été développées dans ce sens [CLD+05, HJRS01], et avec conservation de la taille initiale de l'image [FGCP04]. Il existe également de nombreuses méthodes de compression d'images sans perte [PM92, MB01], et de méthode qui utilise de technique réglable de compression d'image conjoint avec transformée Mojette [BPD+05]. Toutefois, très peus méthodes proposent de combiner à la fois une compression, un chiffrement et une IDC sans perte [Aut02]. En fait la plupart des méthodes d'IDC réversibles augmente la taille de l'image originale. Il nous a donc semblé nécessaire de trouver des nouvelles méthodes permettant d'insérer une grande quantité d'information dans l'image sans augmenter la taille de celle-ci et sans en modifier le contenu tout en la chiffrant. Dans cette section, nous développons une méthode réversible combinant cryptage et IDC tout en diminuant la taille de l'image [RPF04, RP06].

La méthode sans perte d'information que nous allons présenter a pour objectif d'avoir en parallèle la compression, l'insertion de données cachées (IDC) et le chiffrement des images numériques [PR04a]. Le plan général de notre approche est présenté figure 4.1.

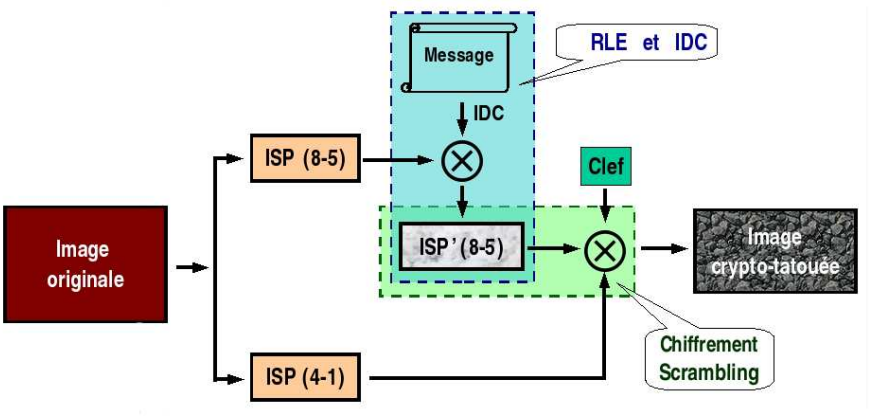


FIG. 4.1 – Plan général de la méthode.

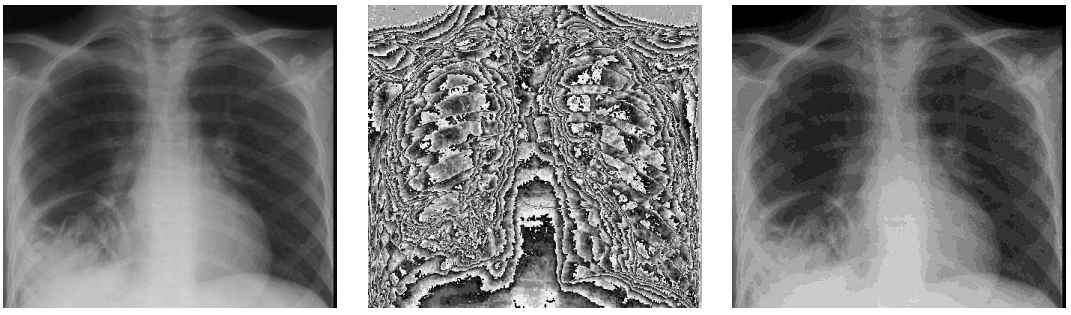
L'idée principale est de décomposer l'image originale en deux images semi-pixels

(ISP) de 4 bits par pixel chacune. Sur chaque image semi-pixels sont employées des procédures spécifiques pour rendre, à la fin, une image chiffrée qui transporte des données cachées. L'image finale est appelée l'image crypto-tatouée. L'image de semi-pixels ISP(8-5) est construite avec les 4 bits de poids les plus forts. L'autre image de semi-pixels, la ISP(4-1), est composée des 4 bits de poids les plus faibles. L'ISP(8-5) est soumise à une compression basée sur le codage par plage pour obtenir de la place afin d'appliquer l'IDC. L'insertion du message dans l'ISP(8-5) est faite par une approche additive. La dernière étape est le chiffrement à l'aide d'une clef secrète et d'un mélange ou brouillage (*scrambling*) avec l'ISP(4-1).

4.1.1 Décomposition d'image

Avant de décrire le procédé de décomposition, il est nécessaire de faire quelques remarques. La systématique employée sur l'image ISP(8-5) est basée sur le travail de Maniccam et Bourbakis [MB01] qui décompose l'image en 8 plans binaires. Ils démontrent que le plus haut taux de compression est accompli dans les plans binaires de bits de poids les plus forts. Notre méthode innove par rapport aux autres [FGCP04, MB01] par l'utilisation de deux plans et de plusieurs techniques concomitantes. Ceci fait gagner en temps de calcul.

La première étape est donc la décomposition de l'image originale en deux images semi-pixels. Le procédé de création des ISPs est très simple et performant parce qu'il est réalisé à l'aide d'une fonction *bitwise*[1] sur le pixel. L'ISP(4-1) est composée des quatre bits de poids les plus faibles, les bits {4, 3, 2, 1} qui représentent les détails de l'image. L'ISP(8-5) est composée des quatre bits restants, les bits de poids les plus forts {8, 7, 6, 5} qui représentent tous les éléments de l'image sans les détails. La figure 4.2 présente la décomposition d'une image médicale en deux images de semi-pixels. Nous pouvons remarquer que l'ISP(8-5) ressemble beaucoup à l'image originale sans les détails.



(a) (b) (c)

FIG. 4.2 – a) Image originale, b) ISP(4-1), c) ISP(8-5).

La figure 4.3 montre les niveaux de gris d'une partie de l'image originale 4.2.a et

[1] Les fonctions bitwise sont des opérateurs (and, or, xor, etc) pour manipuler des bits.

leurs résultats numériques après la décomposition. Nous remarquons qu'avec la décomposition, l'image ISP(8-5), figure 4.3.c, possède beaucoup de semi-pixels identiques. Ceci va rendre efficace la compression par plage.

61	58	58	55	51
51	52	61	53	50
57	56	57	57	55
62	63	62	59	59
49	47	49	50	49

(a)

13	10	10	06	03
03	04	13	05	02
09	08	09	09	07
14	15	14	11	11
01	15	01	02	01

(b)

03	03	03	03	03
03	03	03	03	03
03	03	03	03	03
03	03	03	03	03
03	02	03	03	03

(c)

FIG. 4.3 – a) Partie de l'image originale, b) Même partie dans ISP(4-1), c) Même partie dans ISP(8-5).

4.1.2 Compression sans perte

Pour obtenir des espaces pour l'insertion de données cachées sans augmenter la taille de l'image originale, il est nécessaire de soumettre l'image à une compression. L'ISP(8-5) est compressée en utilisant un codage par plage, RLE (*Run Length Encoding*), présenté chapitre 3, section 3.2.4.

Le codage RLE classique remplace une suite de valeurs identiques par un bloc spécial B. Ce bloc spécial B est composé de (f, q, C), où f est une marque pour identifier le bloc, q est le nombre d'éléments répétés dans la suite et C est la valeur qui se répète.

Nous proposons une variation du RLE. Il s'agit de l'algorithme de codage par plage RLE2IDC (RLE pour l'IDC). Dans le RLE2IDC, le bloc spécial B est constitué de $(f, Q(H, L), C)$, où $Q(H, L)$ est fonction de l'espace H pour l'IDC et de la longueur L de la suite de pixels identiques. Dans notre approche nous avons deux types de blocs B, les blocs B_8 avec 8 bits et les B_{16} avec 16 bits. Pour ces deux types de blocs, B_8 et B_{16}, le nombre de bits utilisé pour la marque identifiant f et pour la valeur C du niveau de gris du pixel sont statiques. La marque f utilise toujours 1 bit et la valeur C toujours 4 bits. Cependant, le nombre de bits pour représenter H et L est variable et dépend l'un de l'autre et du contenu de l'image. Plus H est long, plus L est court.

La figure 4.4 présente la disposition du bloc B_8 et leurs contraintes. La valeur de H_8 correspond à la quantité de bits utilisée pour l'IDC, L_8 est la quantité de bits utilisée pour la longueur de la suite de pixels et le bit Q peut être utilisé comme L ou comme H. Si $Q \in H$ alors tous les blocs B_8 possèdent un bit pour l'IDC. Nous observons, figure 4.4, que les tailles de H et L sont dynamiques et que f et C sont statiques. Nous observons également que dans le bloc B_8 nous pouvons avoir au maximum 1 bit pour l'IDC.

La figure 4.5 présente la disposition du bloc B_{16}, où H_{16} est le nombre de bits utilisé pour l'IDC dans tous les blocs B_{16}, L_{16} est le nombre de bits pour représenter la longueur de la suite de pixels, et Q est variable et peut être employé comme H ou L. Dans le bloc B_{16}, nous avons toujours au moins 2 bits pour l'insertion de données cachées.

	Valeur	Taille en bits
f	0	1
H	$0 \leq H_8 \leq 1$	minimum 0
L	$L_8 = 3 - H_8$	minimum 2
C	couleur	4
	Total	8

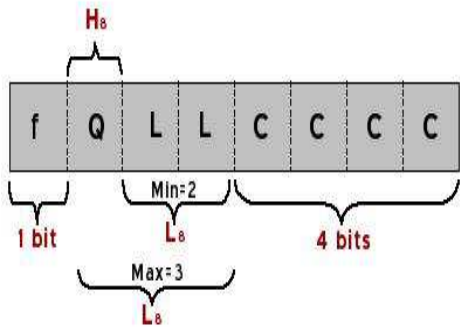


FIG. 4.4 – La représentation du bloc B_8.

	Valeur	Taille en bits
f	1	1
H	$2 \leq H_{16} \leq 7$	minimum 2
L	$L_{16} = 11 - H_{16}$	minimum 4
C	couleur	4
	Total	16

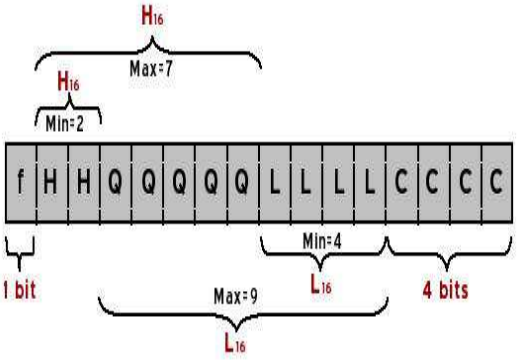
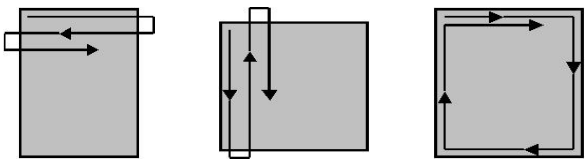


FIG. 4.5 – La représentation du B_{16}.

Parcours du ISP(8-5)

Dans le procédé de codage par plage de notre méthode, l'ISP(8-5) est parcourue par trois chemins différents (par ligne, par colonne et spirale), comme le représente la figure 4.6.



Par ligne (a) | Par colonne (b) | Spirale (c)

FIG. 4.6 – Les trois parcours dans ISP(8-5).

La façon de parcourir l'image semi-pixels ISP(8-5) déterminera la quantité de blocs

B_8 et B_{16}. Ces blocs sont créés en fonction de la longueur L de la suite traitée. Si L peut être représentée par 3 bits (longueur de 1 à 8), nous utilisons un bloc B_8. Si L est plus long (9 à 512), nous employons alors un bloc B_{16}. Si l'image originale possède des grandes zones homogènes, alors le nombre de pixels identiques est important. Notre algorithme produit donc beaucoup de blocs B_{16}. Si les zones homogènes parcourues sont de tailles relativement petites, l'algorithme opte pour des blocs B_8.

Il existe plusieurs façons de parcourir une image. Maniccam et Bourbakis [MB01] ont présenté une méthode pour la compression sans perte basée sur divers chemins de balayage (SCANs). Dans leur travail, l'image est divisée en blocs carrés ($2^k \times 2^k$, $k \geqslant 2$) et chaque bloc est balayé au minimum par 32 modèles de chemins différents. Leur méthode produit un taux de compression efficace, cependant il est très coûteux en temps de calcul.

Le but principal de notre travail n'est pas de trouver le meilleur chemin pour obtenir un taux de compression élevé, mais de trouver une compression raisonnable dans un temps efficace. Ainsi, nous utilisons seulement les trois chemins de balayage présentés figure 4.6. Notre but est de trouver le chemin le plus efficace, une capacité d'IDC élevée et une taille de l'image finale réduite avec un faible temps de calcul.

Longueur maximale de L

Comme montré figures 4.4 et 4.5, l'élément Q dans les blocs B_8 et B_{16} peut être utilisé pour L ou par H. Dans le cas de l'utilisation de Q pour L, nous bénéficions d'une meilleure compression. Si Q est utilisé pour H, alors nous bénéficions d'une plus grande capacité pour l'IDC. Avant de mettre en route le procédé d'insertion de données cachées, décrit dans la section suivante, nous devons définir les LML (longueur maximale de L) pour L_8 et L_{16} fonction du taux de compression souhaité et fonction de la capacité de l'IDC souhaitée.

Pour définir ces longueurs maximales, l'image est parcourue en considérant que la longueur maximale de L_8 pour les blocs type B_8 ne peut être que { 4 ou 8 }, (voir le tableau 4.1). C'est-à-dire, si nous estimons L à 4, nous avons besoin de 2 bits pour représenter la valeur 4 [2] et il nous reste encore 1 bit dans B_8 pour l'IDC. Si nous estimons à 8 la longueur maximale de L_8, alors nous avons besoin de 3 bits pour représenter cette valeur et il reste aucun bit pour l'IDC, comme présenté dans le tableau 4.1.(a). Pour les blocs type B_{16}, nous parcourons l'image en considérant que la longueur maximale de L_{16} peut être { 16, 32, 64, 128, 256 ou 512 bits}. Ensuite, nous utilisons le même raisonnement employé pour le bloc type B_8 expliqué ci-dessus. Le tableau 4.1.(b) montre que nous pouvons avoir jusqu'à 7 bits pour l'IDC. Nous remarquons que dans les blocs B_{16} nous avons au minimum 2 bits pour l'IDC.

La LML est déterminée de façon dynamique pour chaque type de blocs B_8 et B_{16}. Elle est définie en fonction de la taille finale de l'image et de l'espace total W réservé à l'IDC dans l'image. L'équation (4.1) présente l'espace total W disponible pour l'IDC.

$$W = n_8.H_8 + n_{16}.H_{16} + 8.\Delta, \tag{4.1}$$

[2] Les valeurs codées dans 2 bits sont 1, 2, 3, 4.

B_8		
L_8 maximale	4	8
Bits pour l'IDC	1	0

(a)

B_{16}						
L_{16} maximale	16	32	64	128	256	512
Bits pour l'IDC	5+2	4+2	3+2	2+2	1+2	0+2

(b)

TAB. 4.1 – Longueur maximale pour L_8 et L_{16}.

où n_8 et n_{16} sont les nombres de blocs B_8 et B_{16} respectivement, H_8 et H_{16} l'espace réservé à l'IDC et Δ est le reste de la division de la taille de l'image ISP(8-5) comprimée par le nombre de colonnes de l'image originale. En effet, nous supposons que l'image originale et l'image finale comprimée possèdent la même largeur.

Les longueurs L_8 et L_{16}, sont alors considerées comme maximales pour les blocs B_8 et B_{16}.

Fonctionnement de l'algorithme de compression

L'algorithme fonctionne comme suit : l'ISP(8-5) est parcourue suivant les trois parcours proposés (par ligne, par colonne et spirale) en considérant que la plus longue suite pour le bloc type B_8 peut être {4 ou 8} et que pour le bloc B_{16} peut être {16, 32, 64, 128, 256 ou 512}. Ensuite, la LML est définie pour les blocs B_8 et B_{16}, selon l'objectif principal désiré, compression ou IDC. En fin de l'algorithme de compression, l'ISP(8-5) est réduite à l'ISP'(8-5).

4.1.3 Insertion de données cachées

Une fois que l'algorithme de RLE2IDC a été exécuté et que nous avons la grandeur W de l'espace total disponible pour l'IDC, nous pouvons procéder au marquage additif, présenté figure 4.7, pour insérer le message. L'IDC est effectuée dans l'ISP'(8-5).

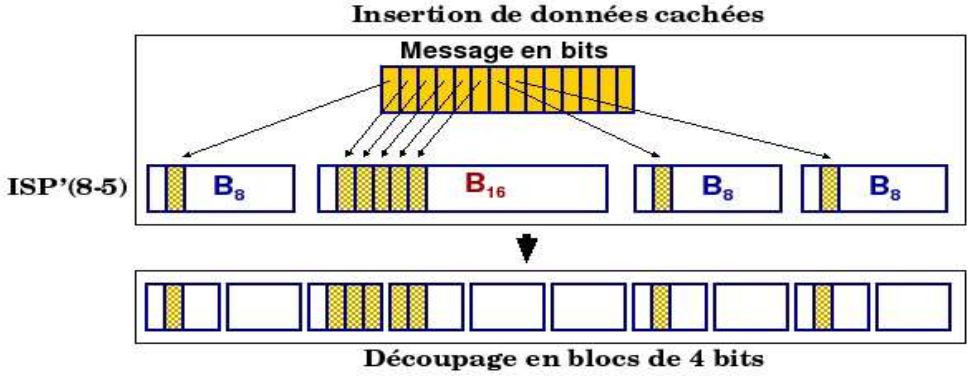


FIG. 4.7 – Schéma d'insertion de données cachées et découpage en blocs de 4 bits.

L'insertion additive du message est faite bit par bit. Le message est coupé en morceaux selon l'espace réservé pour l'IDC et stocké dans les blocs B_8 et B_{16}, comme le montre la figure 4.7. Le vecteur avec le résultat de l'IDC est alors découpé en blocs de 4 bits par pixel. Ces blocs de 4 bits seront employées dans le procédé de brouillage avec l'image semi-pixels ISP(4-1), qui possède aussi 4 bits par pixel. Il est important de remarquer qu'avec ce procédé de découpage les informations vont être dispersées dans l'image. Nous pouvons remarquer que les contenus de ces morceaux de 4 bits sont bien diversifiés. Ils peuvent être composés de fHLL, fLLL, fHHH, fHHL, HHHH, LLLL, CCCC, ..., (voir les figures 4.4 et 4.5).

4.1.4 Brouillage et cryptage sélectif

Après l'IDC nous faisons un brouillage et un cryptage sélectif. Il existe deux approches permetant d'assurer la confidentialité : le cryptage total de l'information et le cryptage sélectif (CS). Le CS assure la confidentialité d'une façon plus adaptée au niveau de protection et en fonction de l'application et du temps disponible. Le cryptage sélectif peut correspondre à des applications qui ne nécessitent pas un cryptage complet mais uniquement un cryptage des données essentielles et pertinentes. Cependant, la sécurité d'un CS est toujours plus faible comparée à celle d'un cryptage complet. La seule raison d'accepter ce schéma est la réduction importante du temps de calcul par rapport à un cryptage total.

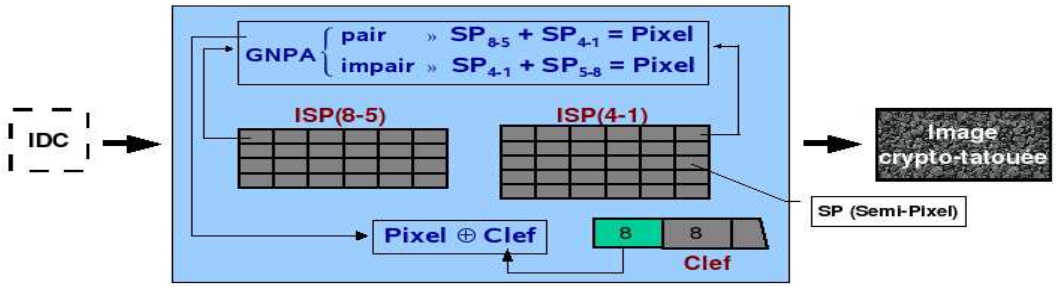


FIG. 4.8 – Brouillage / cryptage sélectif.

Le processus de cryptage sélectif de notre méthode (figure 4.8) est basé sur deux étapes : le brouillage et la fusion. Le brouillage/fusion groupe les deux semi-images ISP(4-1) et ISP'(8-5). Même si le procédé de masquage de données diffuse l'information sur l'ISP'(8-5) d'une façon inéligible, il ne peut pas être considéré comme une méthode de chiffrement. Une méthode de chiffrement qui dépend du secret de l'algorithme de cryptage n'est pas vraiment une méthode de cryptage. La securité de la méthode de cryptage proposée repose sur une clef secrète. La clef est divisée en sous-clefs de 8 bits chacune. Ces sous-clefs de 8 bits sont combinées par une opération d'un ou exclusif avec les pixels obtenus de la fusion de l'ISP'(8-5) et de l'ISP(4-1). La clef est encore employée comme semence d'un générateur de nombre pseudo-aléatoire (GNPA). Ce GNPA produit des nombres qui seront utilisés dans les trois parties suivantes du processus :

1. Pour indiquer, dans l'image finale (crypto-tatouée), la localisation des informations critiques qui sont nécessaires pour le procédé d'extraction.
2. Pour sélectionner le bon semi-pixels dans l'ISP(4-1) au moment de la fusion.
3. Pour décider l'ordre de la fusion ISP(8-5) → ISP(4-1) ou ISP(4-1) → ISP(8-5).

Le premier semi-pixels de l'ISP'(8-5) est combiné avec le semi-pixels de l'ISP(4-1) obtenu à travers du GNPA pour produire un octet. Nous remarquons que le tout premier nombre pseudo-aléatoire produit indique la localisation du vecteur des informations critiques. Ce premier nombre n'est donc pas utilisé pour la fusion. Il est important de signaler que les GNPA peuvent répéter un nombre. Le vecteur de colision est un mécanisme utilisé pour éviter l'utilisation d'un nombre déjà employé. Dans notre approche, nous utilisons un vecteur de colision de la taille de l'ISP(4-1). Chaque fois qu'un nombre pseudo-aléatoire est généré et utilisé pour la fusion, il est indiqué dans le vecteur de colision.

Si le nombre pseudo-aléatoire généré est pair, la fusion est faite dans l'ordre ISP(8-5) → ISP(4-1), sinon ISP(4-1) → ISP(8-5). Le brouillage des données est très important dans un système de cryptage. L'algorithme standard de cryptage AES [DR02b] mélange également des données à travers des fonctions *ShiftRows* et *MixColumns*. Ces fonctions sont des mécanismes facilement implantés en processeurs de 8 bits. Dans notre algorithme de cryptage sélectif, nous employons une partie de la clef dans une opération d'un ou exclusif avec le pixel précédemment fusionné, comme présenté sur la figure 4.8. La motivation d'utilisation de ce mélange est la résistance contre les attaques carrées et différentielles [WLW03].

Comme la taille de l'ISP'(8-5) est plus petite que la taille de l'ISP(4-1) (car comprimée), ce processus est fait jusqu'à la fin de l'ISP'(8-5). Les valeurs restantes dans l'ISP(4-1) seront fusionnées entre elles d'une façon aléatoire toujours à l'aide du vecteur de colision.

Informations critiques pour le décodage

Le procédé de décodage, présenté section 4.1.5, nécessite quelques informations qui doivent être insérées dans l'image crypto-tatouée. Nous allons donc construire le VIC (Vecteur d'Informations Critiques) avec 8 octets. Ce vecteur possède les informations nécessaires pour le décodage. Le tableau 4.2 présente la composition du VIC. Ce vecteur est inséré dans l'image crypto-tatouée à l'aide du premier nombre produit par le GNPA.

4.1.5 Décodage

Le procédé de décodage est composé du décryptage, de l'extraction du message et de la reconstruction de l'image originale. Pour le récepteur qui possède l'image crypto-tatouée et la clef, le procédé de décryptage jusqu'à la reconstruction de l'image originale est très simple et efficace.

Nombre de bits	Description
1	pour définir le nombre de bits utilisé pour l'IDC dans les blocs H_8
3	pour définir le nombre de bits utilisé pour l'IDC dans les blocs H_{16}
2	pour le mode de balayage (ligne, colonne, spirale)
12	pour le nombre de colonnes de l'image originale
12	pour le nombre de lignes de l'image originale
16	pour la taille, en octet, du message inséré
18	pour la taille de l'ISP'(8-5)

TAB. 4.2 – Les données critiques pour l'algorithme de décodage.

Récupération des informations critiques

La clef secrète est employée comme semence pour le GNPA. Le premier nombre produit indique la localisation du VIC dans l'image crypto-tatouée. A partir de ce vecteur, nous obtenons toutes les informations nécessaires pour les phases suivantes, comme le type de parcours et la taille de l'ISP'(8-5), par exemple.

Décryptage

L'image crypto-tatouée est parcourue en prenant un octet à la fois. Cet octet est décomposé en deux morceaux de 4 bits chacun concernant la parité du nombre aléatoire généré. Ces morceaux vont construire l'ISP'(8-5) et l'ISP(4-1) selon le GNPA. Cette procédure est faite jusqu'à obtenir la taille de l'ISP'(8-5) précisée dans le VIC. Après la construction de l'image l'ISP'(8-5), l'ISP(4-1) est déduite avec les informations restantes dans l'image crypto-tatouée.

L'extraction du message

L'extraction du message est faite à partir de l'ISP'(8-5) déjà construite dans l'étape précédente. Pour l'extraction du message tout d'abord, nous obtenons dans le vecteur VIC, les nombres de bits utilisés pour l'IDC dans chaque bloc B_8 et B_{16}. Avec ces informations, nous pouvons donc parcourir l'ISP'(8-5), extraire les bits H_8 et H_{16} utilisés pour l'IDC et construire le message.

Décompression

Après l'extraction du message, il est nécessaire de faire la décompression. Il s'agit de lire le premier bit de l'octet de l'ISP'(8-5) pour obtenir le type de bloc. Avec le type de bloc, nous avons la longueur L de la suite et la couleur C. Il suffit alors de construire la suite de L pixels identiques avec la couleur C.

Fusion de l'ISP(8-5) et l'ISP(4-1)

L'ISP(8-5) a été générée dans l'étape précédente. Il suffit d'utiliser l'équation (4.2) composée des opérateurs *bitwise* pour fusionner l'ISP(8-5) avec l'ISP(4-1) et générer l'image originale :

$$Pixel_{Original} = SemiPixel_{(8-5)} << 4 \quad | \quad SemiPixel_{(4-1)}. \tag{4.2}$$

4.1.6 Résultats

Après avoir décrit en détail notre méthode, nous présentons les résultats obtenus sur des images réelles. La méthode a été appliquée sur 35 images en niveau de gris. Elles ont été chiffrées avec une clef de 128 bits et tous les espaces W pour le marquage ont été remplis par un message. Nous présentons les résultats détaillés pour deux images et un tableau final résume les résultats moyens de toutes les images. Nous avons appliqué l'algorithme de décodage sur les images et nous avons réussi à extraire les messages et à reconstruire les images originales sans aucune perte.

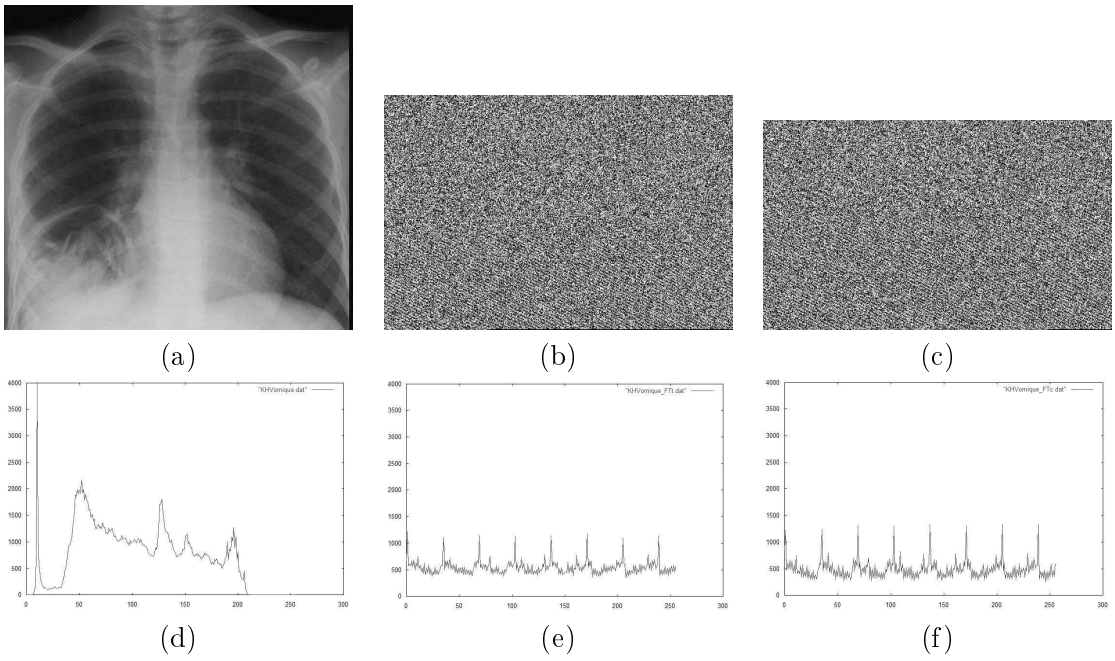


FIG. 4.9 – a) Image originale KHVomique (446 x 415), b) Image crypto-tatouée avec une capacité maximale W, c) Image crypto-tatouée avec une compression maximale, d) Histogramme de l'image originale, e) Histogramme de l'image (b), f) Histogramme de l'image (c).

Le tableau 4.3 présente les résultats pour les images médicales KHVomique illustrée figure 4.9 et Colposcopie figure 4.10.

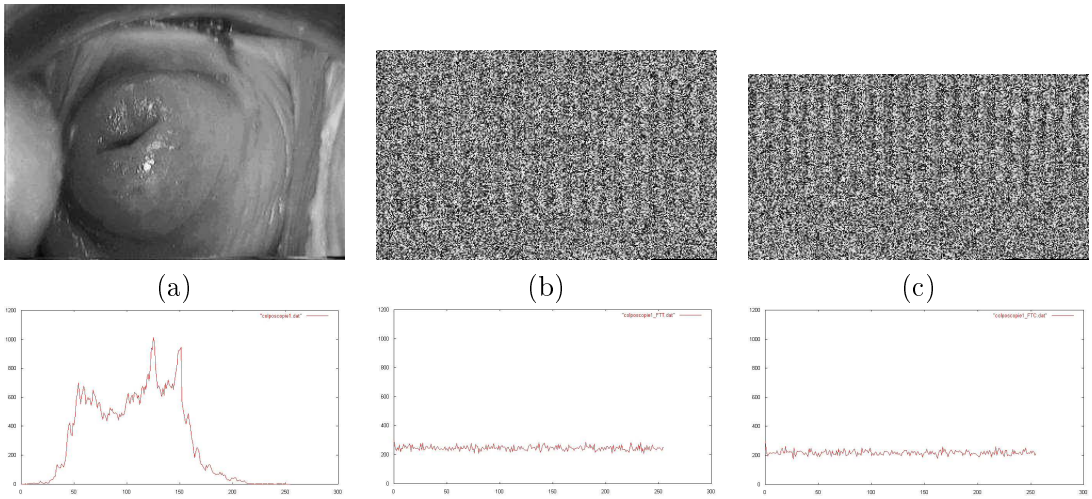


FIG. 4.10 – a) Image originale colposcopie (446 x 415), b) Image crypto-tatouée avec une capacité maximale W, c) Image crypto-tatouée avec une compression maximale, d) Histogramme de l'image originale, e) Histogramme de l'image (b), f) Histogramme de l'image (c).

Image	KHVomique		Colposcopie	
Taille originale (octets)	185090		76800	
Optimal	Taille de W	Compression	Taille de W	Compression
Type parcours	ligne	spirale	ligne	cologne
H_8 | H_{16}	1 | 7	0 | 2	1 | 7	0 | 4
n_8	11301	12467	13552	12452
n_{16}	14598	6994	5353	2375
Taille finale (octets)	133131	119305	62720	55680
Compression τ	1,39	1,55	1,22	1,38
W (octets)	14275	2054	6440	1266
W pourcentage	7,71	1,11	10,27	2,27
Entropie (bits/pixel)	7,96	7,93	7,99	7,99

TAB. 4.3 – Résultat sur des images médicales

La figure 4.9.a présente l'image médicale (KHVomique). L'application de notre méthode en cherchant un **espace optimal pour l'IDC** nous permet d'obtenir $W = 14275$ octets qui correspondant à 7,71 % de l'image originale. Avec cet espace pour l'IDC, nous pouvons insérer un rapport médical de 8 pages, l'historique complet du patient. En plus d'avoir insérer cette quantité d'information, la taille de l'image finale (133131 octets) est 28,07% plus petite que l'image originale (185090 octets). L'application de notre méthode pour un **taux de compression optimal**, nous permet d'obtenir une image crypto-tatouée avec une taille finale 35,54 % plus petite que l'image originale. L'espace pour l'IDC obtenu pour une compression maximale est de 2054 octets. Cet espace est suffisant pour deux pages de type A4 pour le diagnostic du patient.

La figure 4.10 présente l'image médicale (colposcopie). L'application de notre méthode pour un espace optimal pour l'IDC nous permet d'obtenir $W = 6440$ octets correspondant à 10,27% de la taille de l'image originale. Avec cet espace pour l'IDC, nous pouvons insérer un rapport médical de 5 pages d'informations du patient. La taille de l'image finale (62720 octets) est 18,33% plus petite que l'image originale (76800 octets). Pour un taux de compression optimal de 1,38 notre méthode nous permet d'obtenir une image crypto-tatouée avec une taille finale 27,50 % plus petite que l'image originale. Dans cette image compressée, nous avons un espace pour l'IDC de 1266 octets. Cet espace est suffisant assez pour une page de type A4 de diagnostic du patient.

Le tableau 4.4 présente les résultats (valeurs moyennes) sur 35 images. Pour **l'IDC optimal** nous avons en moyenne 8,21% de la taille de l'image originale et même avec cette capacité, notre méthode obtient un taux de compression de 1,32. Pour **la compression maximale** nous avons un taux de compression moyen de 1,55 et une capacité pour l'IDC de 1,13 % de l'image originale. Pour les deux cas, les entropies sont égales à 7,98 bits/pixel.

	Capacité %	Taux de Compression	Entropie bits/pixel
IDC maximale	8,21	1,32	7,98
Compression mximale	1,13	1,55	7,97

TAB. 4.4 – Résultat moyen issus de 35 images médicales.

4.1.7 Bilan

Dans cette section, nous avons présenté une méthode réversible qui combine les trois domaines le chiffrement, l'IDC et la compression. L'originalité de notre méthode est à la fois une compression, un chiffrement et une IDC sans perte. Cette méthode est basée sur la décomposition de l'image, sur une compression sans perte s'appuyant sur de l'algorithme RLE et sur la définition de la LML.

La méthode proposée est capable de cacher une information de taille égale à 8% de la taille de l'image originale. Le taux de compression et la capacité d'insertion sont dynamiques. La méthode de chiffrement est basée sur des opérations de l'algorithme

AES. Nous avons atteint une entropie très proche de 8 bits/pixel. La capacité d'IDC proposée est significativement plus importante qu'avec les méthodes classiques d'IDC dans le domaine spatial. Cette méthode propose un algorithme de chiffrement sélectif efficace et permet de reconstruire l'image originale sans aucune perte. Dans notre méthode, trois processus, l'IDC, la compression et le cryptage ont été groupés en un seul algorithme. Par conséquent cela diminue le temps de traitement et notre méthode est donc applicable sur des systèmes de faibles puissances comme les téléphones mobiles par exemple.

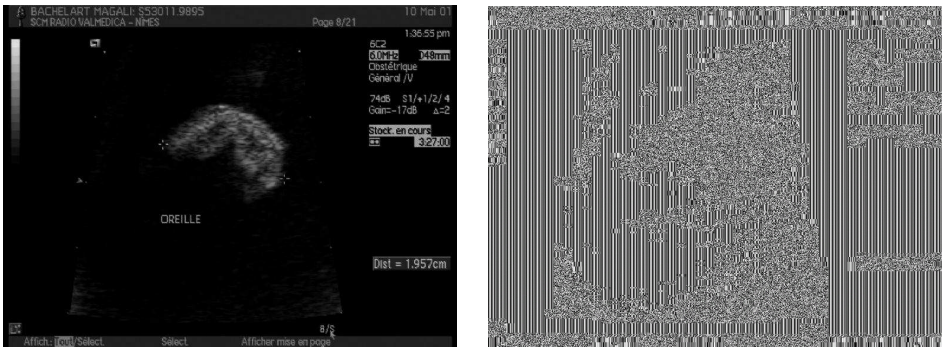
L'inconvénient de cette méthode est la création d'un nouveau format, donc non compatible avec les formats standards d'images.

4.2 Une méthode autonome de masquage de données

Introduction

Le transfert d'images augmente de plus en plus sur Internet et la sécurité des transferts devient très importante. Il existe deux possibilités pour protéger des données durant leur transmission. La première possibilité consiste à crypter les images [CHC01, SS03]. Dans ce cas une clef est nécessaire pour décrypter l'image à la réception. La seconde possibilité consiste à utiliser des méthodes d'IDC. Ces deux approches sont complémentaires pour la protection. Dans cette section nous proposons une nouvelle méthode [PR06] de protection combinant plusieurs catégories de cryptage (symétrique, asymétrique et par flot asynchrone) d'images et l'IDC.

Nous avons vu, chapitre 1, que les processus de cryptage pouvaient être symétriques ou asymétriques par bloc ou par flot. Nous avons vu que les algorithmes asymétriques ont des temps de calcul très important et par conséquent ils ne sont pas appropriés au cryptage des images. Les algorithmes par bloc présentent trois inconvénients lorsqu'ils sont appliqués aux images. Le premier est quand nous avons des zones homogènes alors tous les blocs identiques sont cryptés de la même manière, comme montré figure 4.11.



(a) (b)

FIG. 4.11 – a) Image médicale échographique, avec de grandes zones homogènes, b) Image médicale cryptée par bloc avec AES mode ECB.

Le second inconvénient est que les algorithmes de chiffrement par bloc ne sont pas robustes au bruit. Le troisième problème concerne l'intégrité des données. Les images de la figure 4.12 illustrent ce problème. A partir de l'image originale, figure 4.12.a, nous avons appliqué l'algorithme AES par bloc (mode ECB) avec une clef de 128 bits afin d'obtenir l'image cryptée figure 4.12.b. Si l'image cryptée est modifiée durant le transfert il n'est pas forcément possible de détecter la modification. Par exemple, dans la figure 4.12.c nous avons permuté les quatre régions (modulo 128 bits) de l'image et dans la figure 4.12.d copié une petite région de l'image cryptée et nous avons collé cette région sur une autre zone de l'image. Après décryptage, il est possible de visualiser les images mais il n'est pas possible de garantir l'intégrité comme illustrée figures 4.12.e et f.

De manière générale les méthodes de chiffrement par flot sont plus robustes au

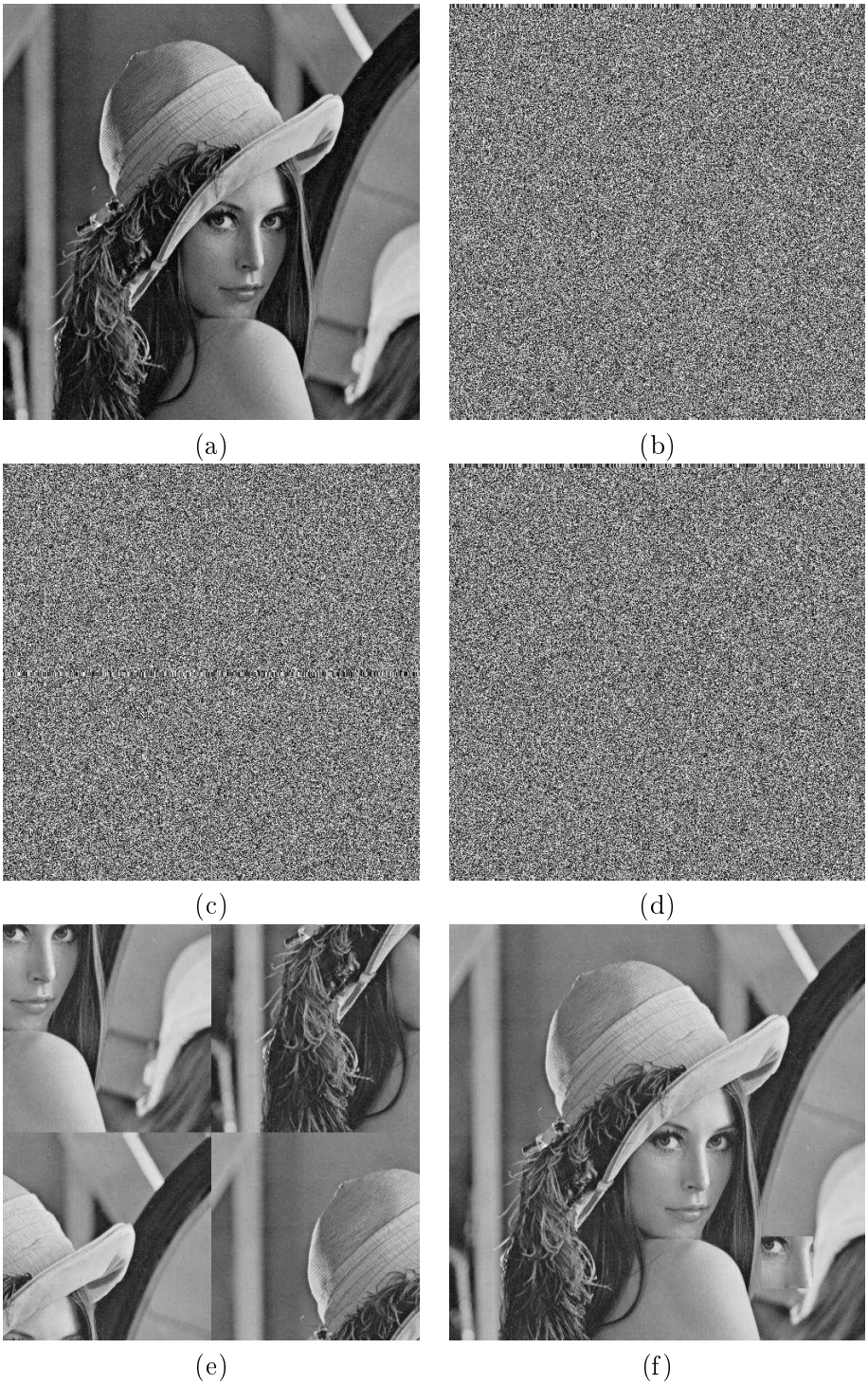


FIG. 4.12 – a) Image originale de Lena, b) Image cryptée avec AES par bloc de 128 bits, c) Permutation de régions de l'image cryptée, d) Copie d'une région de l'image cryptée et collage sur une autre zone, e) Décryptage de (c), f) Décryptage de (d).

bruit que les méthodes par bloc. Pour être robuste au bruit afin d'insérer un message dans l'image cryptée, nous avons choisi d'utiliser l'algorithme de chiffrement par flot asynchrone présenté section suivante.

La sécurité des algorithmes d'IDC et de cryptage existants sont basés sur des **clefs secrètes**. Dans l'approche traditionnelle, l'image est cryptée avec une méthode à clef secrète et la clef secrète est cryptée avec une méthode asymétrique à **clefs publique** et **privée**. Le problème de cette approche est de transférer en même temps l'image cryptée et la clef secrète cryptée. Dans cette section nous présentons un algorithme de chiffrement symétrique d'images pour un transfert sécurisé sans avoir besoin de transférer par un autre canal la clef secrète [PR04b, PR04a]. Pour cela, nous proposons de chiffrer la clef secrète avec un algorithme asymétrique et d'insérer cette clef cryptée dans l'image par IDC. Le fait d'insérer la clef dans l'image rend la méthode autonome et garantit l'intégrité des données. En effet, si l'image est attaquée durant le transfert (par modification de pixels ou découpage par exemple) alors le récepteur n'est plus capable d'extraire de l'image la clef cachée. Et par conséquent il n'est plus possible de décrypter l'image, et donc le problème d'integrité est détecté.

4.2.1 Nouvelle méthode combinant cryptage et IDC

Dans cette section nous proposons une nouvelle méthode combinant un algorithme de chiffrement par flot asynchrone, présenté section 4.2.2, pour chiffrer l'image et un algorithme asymétrique pour chiffrer la clef secrète. Ensuite, nous utilisons une méthode d'IDC pour insérer la clef cryptée dans l'image cryptée. D'un point de vue pratique, si une personne A envoie par réseau une image à B, l'émetteur A utilisera l'algorithme de chiffrement par flot avec la clef secrète K pour crypter l'image. Ensuite, le problème est pour transmettre la clef K. Afin de transmettre cette clef A peut chiffrer la clef K en utilisant un algorithme à clef publique tel que RSA par exemple. Soit $pub\ (b, n)$ la clef publique et $priv\ (v, n)$ la clef privée, alors A a ses clefs publique et privée $pub_A\ (b_X, n_X)$ et $priv_A\ (v_X, n_X)$, et B ses clefs publique et privée $pub_B\ (b_Y, n_Y)$ et $priv_B\ (v_Y, n_Y)$.

Par conséquent A génère une clef secrète K pour cette session et chiffre l'image avec notre algorithme de chiffrement par flot. Ensuite, A signe la clef K avec l'algorithme RSA en utilisant sa clef privée $priv_A$ afin d'obtenir une clef signée K' telle que :

$$K' = K^{v_X} \bmod (n_X). \tag{4.3}$$

Cette clef cryptée K' est cryptée une seconde fois avec RSA en utilisant la clef publique pub_B de son correspondant B afin de générer K'' :

$$K'' = K'^{b_Y} \bmod (n_Y). \tag{4.4}$$

Dans notre méthode de combinaison la taille du message à insérer dans l'image dépend de la taille de la clef publique du récepteur. Cette taille est connue par l'émetteur A et le récepteur B. Nous pouvons donc calculer le facteur d'insertion et calculer le nombre de blocs nécessaires pour la méthode d'IDC. Cette clef K'' est donc le message à insérer dans l'image chiffrée en utilisant la méthode d'IDC. Finalement, A envoie

l'image à B comme présentée figure 4.13. Cette procédure de cryptage K avec $priv_A$ et pub_B assure l'authenticité et seul B peut décrypter l'image envoyée. Le fait d'insérer la clef dans l'image rend la méthode autonome et garantit l'intégrité. En effet, si durant le transfert l'image est attaquée alors il n'est plus possible à la réception d'extraire la bonne clef et donc de décrypter l'image.

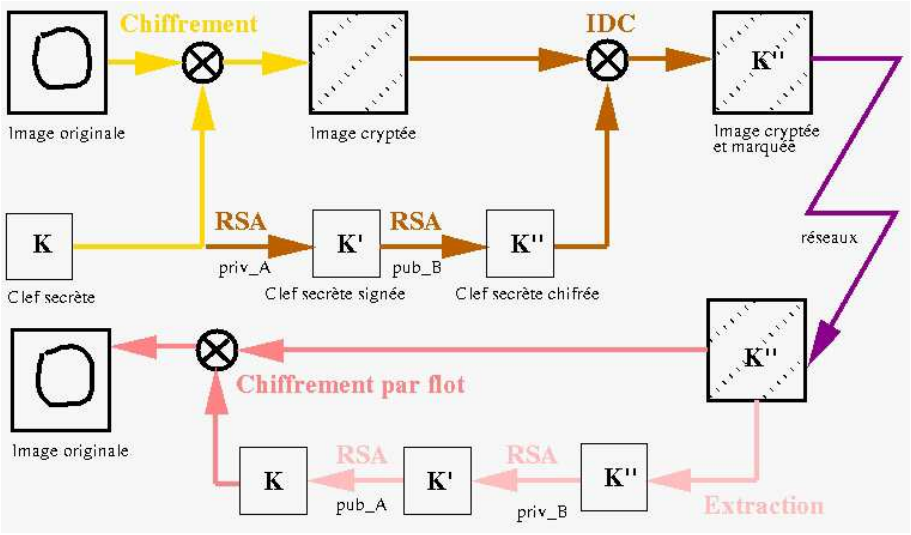


FIG. 4.13 – Combinaison d'un cryptage à clef secrète, d'un cryptage à clef publique et d'une méthode d'IDC.

La personne B reçoit l'image cryptée et marquée et peut alors extraire la clef cryptée K''. Elle peut alors décrypter la clef K'' en utilisant sa clef privée $priv_B$ et authentifier A en utilisant la clef publique pub_A de A telles que :

$$K = (K''^{v_Y} \mod n_Y)^{b_X} \mod (n_X). \tag{4.5}$$

Avec la clef obtenue K, B peut déchiffrer l'image et la visualiser. Si B veut envoyer une nouvelle image à A, il doit générer une nouvelle clef secrète K_1 pour cette nouvelle session. Le processus sera le même mais les clefs publique et privée pour RSA ne seront pas appliquées dans le même ordre. Même si cinq clefs sont nécessaires pour chaque session, celles-ci sont transparentes pour les utilisateurs. En effet les clefs privées sont gérées par les PKI (Infrastructure de gestion de clefs), voir section 1.2.1, et il n'est pas nécessaire de connaître la valeur de la clef secrète qui est cryptée et insérée dans l'image. Toutefois, pour chaque session la valeur de la clef secrète K doit changer. Sinon, si la clef était toujours la même toutes les personnes possédant le logiciel pourraient décrypter les images.

4.2.2 Algorithme de chiffrement par flot asynchrone proposé

Les algorithmes de chiffrement par flot sont composés de deux étapes : la génération d'une clef dynamique et la fonction de cryptage de sortie dépendant de la clef dyna-

mique. Quand la clef dynamique est générée à partir de la clef et d'un certain nombre de digits précédemment crypté, l'algorithme de chiffrement par flot est dit asynchrone.

Les processus de cryptage et décryptage d'un chiffrement par flot asynchrone sont décrit figure 4.14, où $g()$ est la fonction génératrice de la clef dynamique et $h()$ la fonction de sortie de cryptage :

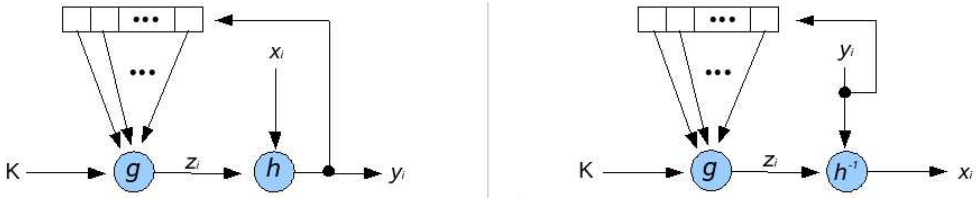


FIG. 4.14 – Cryptage et décryptage asynchrone.

$$\begin{cases} z_\imath &= g(K, c_{\imath-t}, c_{\imath-t+1}, \ldots, c_{\imath-2}, c_{\imath-1}) \\ c_\imath &= h(z_\imath, m_\imath), \end{cases} \tag{4.6}$$

où K est la clef, $m_\imath$, $c_\imath$ et $z_\imath$ sont respectivement le $\imath^{eme}$ texte clair, le texte chiffré et la clef dynamique. Nous pouvons remarquer équations (4.6) que la clef dynamique dépend des t digits précédents du texte chiffré.

Dans cette section nous présentons un nouvel algorithme de chiffrement par flot asynchrone appliqué aux images. Soit K une clef de longueur k bits $b_\imath$ et $K = b_1 b_2 ... b_k$. L'unité de cryptage est le pixel (1 octet). La méthode réside dans le fait que pour chaque pixel de l'image le cryptage dépend du pixel original, de la valeur de la clef K, et des $k/2$ pixels précédemment cryptés. Pour utiliser les équations (4.6) nous avons $t = k/2$.

Pour chaque pixel $p_\imath$ de l'image originale, nous calculons la valeur du pixel $p'_\imath$ de l'image chiffrée en utilisant l'équation suivante :

$$\begin{cases} z_\imath &= (\sum_{\jmath=1}^{k/2} \alpha_\jmath p'_{\imath-\jmath}) \mod 256 \\ p'_\imath &= (z_\imath + p_\imath) \mod 256. \end{cases} \tag{4.7}$$

avec $\imath \in [0, \ldots, N-1]$, où N est le nombre de pixels de l'image, k est la longueur de la clef, avec $k \in [1, N]$, et $\alpha_\jmath$ est une séquence de $k/2$ coefficients générée à partir de la clef secrète K.

Le principe de chiffrement est le même que celui illustré figure 4.14. Les équations (4.7) ont une récurrence d'ordre $k/2$, correspondant à la moitié de la longueur de la clef. Les coefficients $\alpha_\jmath$ sont des coefficients entiers compris entre -2 et $+2$ tels que :

$$\begin{cases} \alpha_\jmath = \beta_\jmath - 1 & \text{si } \beta_\jmath \in \{0, 1, 2\}, \\ \alpha_\jmath = \pm 2 & \text{si } \beta_\jmath = 3, \end{cases} \tag{4.8}$$

avec $\beta_\jmath = 2b_{2\jmath-1} + b_{2\jmath}$, où $b_{2\jmath-1}$ et $b_{2\jmath}$ sont deux bits voisins de la clef secrète K.

De plus, la densité de probabilité des $\alpha_\jmath$ doit être uniforme afin d'atténuer le bruit durant l'étape de décryptage. Le signe devant les coefficients égaux à 2 dépend de la somme des coefficients $\alpha_\jmath$ afin d'avoir :

$$\frac{1}{k/2}\sum_{\jmath=1}^{k/2}\alpha_\jmath \simeq 0. \tag{4.9}$$

Une autre information est également construite à partir de la clef K. En effet, en prenant en compte que le chiffrement d'un pixel s'appuie sur les $k/2$ pixels précédemment cryptés, nous ne pouvons pas chiffrer les $k/2$ premiers pixels de l'image de la même manière. Il est nécessaire d'associer la séquence des coefficients $\alpha_\imath$ à une séquence de $k/2$ pixels virtuels cryptés $p'_{-\imath}$, pour $\imath \in [1, \ldots, k/2]$, correspondant à un vecteur d'initialisation (VI). Par conséquent, un VI est codé dans la clef : $k/2$ valeurs de pixels virtuels qui permettent de crypter les $k/2$ premiers pixels de l'image comme si ils avaient des prédécesseurs.

La longueur k de la clef K doit être suffisamment grande afin de garantir une sécurité maximale. Supposons $k = 128$, comme nous avons 2 bits par coefficient α, l'ordre de la récurrence est 64. Concernant le VI, nous avons montré qu'il était nécessaire pour chiffrer les $k/2$ premiers pixels. Nous ne lui donnons donc pas plus de place supplémentaire dans la clef, mais la valeur du VI est déduite de la clef de 128 bits par le principe suivant. Il est basé sur une fenêtre glissante qui lit les bits de la clef de la gauche vers la droite. La fenêtre lit le premier octet afin de générer le premier pixel virtuel, le système se déplace alors d'un bit de la clef vers la droite afin d'obtenir un nouvel octet et de générer un autre pixel virtuel. Le déplacement de un bit vers la droite s'effectue jusqu'à obtenir le nombre nécessaire de pixels virtuels.

L'équation (4.10) présente la procédure de décryptage. Dans la procédure de décryptage, nous devons appliquer le processus inverse. Nous pouvons noter que la fonction génératrice de la clef dynamique est la même qu'à l'équation (4.7) :

$$\begin{cases} z_\imath &= (\sum_{\jmath=1}^{k/2}\alpha_\jmath p'_{\imath-\jmath}) \mod 256 \\ p_\imath &= (p'_\imath - z_\imath) \mod 256, \end{cases} \tag{4.10}$$

Analyse de la robustesse au bruit de la méthode de chiffrement par flot asynchrone

Dans cette section nous allons analyser la robustesse au bruit de la méthode de chiffrement par flot asynchrone présentée section 4.2.2. Avec un bruit additif $n_\imath$, chaque pixel crypté $p'_\imath$ devient $\tilde{p}'_\imath$ tel que :

$$\tilde{p}'_\imath = p'_\imath + n_\imath, \tag{4.11}$$

avec $n_\imath$ un bruit additif simulant le bruit dû à l'IDC, tel que $n_\imath \in \{-1, 1\}$ et $Pr(-1) = Pr(1) = \frac{1}{2}$, où $Pr(x)$ est la probabilité d'avoir x.

Pendant le décryptage, à partir des équations (4.10) et (4.11) nous avons :

$$\tilde{p}_\imath = (\tilde{p}'_\imath - \sum_{\jmath=1}^{k/2} \alpha_\jmath \tilde{p}'_{\imath-\jmath}) \mod 256. \tag{4.12}$$

Supposons deux cas particuliers. Le premier cas est quand nous avons seulement un pixel bruité tous les $k/2$ pixels. Dans ce cas l'EQM' de l'image cryptée bruitée par rapport à l'image cryptée est :

$$\begin{aligned} EQM' &= \tfrac{1}{N} \sum_{\imath=0}^{N-1} (p'_\imath - \tilde{p}'_\imath)^2 \\ &= \tfrac{2}{k}. \end{aligned} \tag{4.13}$$

Par exemple, pour $k = 128$, l'$EQM' = 15,6\ 10^{-3}$, et le $PSNR = 66,19\ dB$. Dans ce premier cas, l'EQM de l'image décryptée est :

$$\begin{aligned} EQM &= \tfrac{1}{N} \sum_{\imath=0}^{N-1} (p_\imath - \tilde{p}_\imath)^2 \\ &= 1. \end{aligned} \tag{4.14}$$

La qualité de l'image décryptée ne dépend donc pas de la longueur k de la clef K, le $PSNR = 48,13\ dB$.

Le second cas particulier est quand tous les pixels cryptés sont bruités. Dans ce cas, l'EQM' de l'image cryptée est :

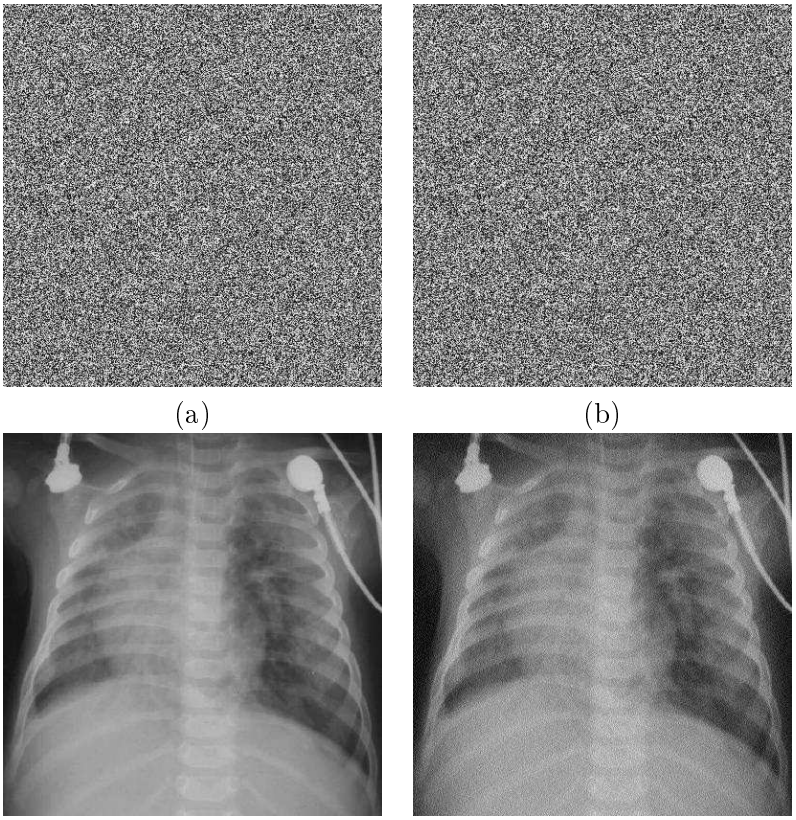
$$\begin{aligned} EQM' &= \tfrac{1}{N} \sum_{\imath=0}^{N-1} (p'_\imath - \tilde{p}'_\imath)^2 \\ &= \tfrac{1}{N} \sum_{\imath=0}^{N-1} n_\imath^2 \\ &= 1. \end{aligned} \tag{4.15}$$

Dans ce second cas la qualité de l'image cryptée ne dépend pas de la longueur k de la clef K, le $PSNR = 48,13\ dB$. Dans ce second cas particulier pour l'image décryptée, l'EQM est :

$$\begin{aligned} EQM &= \tfrac{1}{N} \sum_{\imath=0}^{N-1} (\sum_{\jmath=1}^{k/2} \alpha_\jmath n_\imath)^2 \\ &= 1 + \tfrac{3k}{8}. \end{aligned} \tag{4.16}$$

Cette valeur est obtenue à partir des équations (4.8) et (4.9) en considérant que nous avons $Pr(\alpha_\imath = 0) = \frac{1}{4}$, $Pr(\alpha_\imath = \pm 1) = \frac{1}{2}$ et $Pr_\alpha(\imath = \pm 2) = \frac{1}{4}$, où $Pr(x)$ est la probabilité d'avoir x. Dans ce second cas, avec $k = 128$ pour l'image décryptée le $PSNR = 31,23\ dB$.

En conclusion, dans le premier cas la différence de qualité entre l'image cryptée bruitée et l'image décryptée bruitée est de $18,06\ dB$. Dans le second cas, le bruit est plus intense, la différence de qualité diminue à $16,90\ dB$. Nous pouvons donc conclure que si il y a du bruit, quelque soit son intensité nous perdons plus de $16\ dB$ de qualité. Mais même dans le second cas la qualité de l'image finale reste supérieure à $30\ dB$.



(a) (b)

(c) (d)

FIG. 4.15 – a) Image cryptée avec l'algorithme de chiffrement par flot et bruitée (TEB=$1,95\ 10^{-3}$), b) Image cryptée avec l'algorithme de chiffrement par flot et bruitée (TEB=$1,25\ 10^{-1}$), c) Résultat du décryptage de l'image figure (a), d) Résultat du décryptage de l'image figure (b).

Afin de vérifier expérimentalement la robustesse au bruit de notre méthode, nous avons ajouté un bruit sur l'image cryptée, figure 4.15.a, avec un taux d'erreur binaire (TEB) de $1,95\ 10^{-3}$, figure 4.15.a. (ce TEB correspond à la modification du LSB de 1 pixel tous les 64 ($k/2$)) et avec un TEB de $1,55\ 10^{-1}$, figure 4.15.b (ce TEB correspond à la modification des LSBs de tous les pixels). Les différences entre l'image cryptée et les deux images cryptées bruitées donnent un $PSNR = 66,15\ dB$ pour le TEB $1,95\ 10^{-3}$ (valeur théorique $PSNR = 66,19\ dB$) et un $PSNR = 48,13\ dB$ pour le TEB de $1,25\ 10^{-1}$ (valeur théorique $PSNR = 48,13\ dB$). Après décryptage des

images cryptées bruitées, nous obtenons les images décryptées illustrées figures 4.15.c et d. Les différences entre l'image originale et les images décryptées bruitées, donne un $PSNR = 46,18\ dB$ (valeur théorique $PSNR = 48,13\ dB$) pour le TEB de $1,95\ 10^{-3}$ et un $PSNR = 27,74\ dB$ (valeur théorique $PSNR = 31,23\ dB$) pour le TEB de $1,25\ 10^{-1}$. Nous avons de petites différences entre les valeurs théoriques et notre exemple en partie à cause de l'équation (4.9) qui n'est pas exactement respectée.

Nous pouvons conclure de cette analyse qu'avec notre méthode de chiffrement par flot il est possible d'ajouter un bruit dans une image cryptée et d'obtenir un décryptage de l'image de bonne qualité. Dans la section suivante, le bruit cité sera l'insertion de la clef cryptée dans l'image.

4.2.3 L'insertion de données cachées dans le domaine spatial

Dans ce travail, nous avons employé un algorithme d'IDC dans le domaine spatial. Les bits du message sont insérés directement dans les LSB (le bit de poids le plus faible) de certains pixels. Ainsi l'objectif est d'inclure un message M, la clef chiffrée. Le message M est composé de k bits b et $M = b_1\, b_2\, b_3 \ldots b_k$. Le schéma d'insertion de données cachées est composé des étapes suivantes :

- Tout d'abord, nous calculons un facteur d'insertion $F_i = 1 : \lfloor N/k \rfloor$ (1 bit du message tous les F_i pixels), où N est le nombre de pixels de l'image et k est le nombre de bits du message à insérer.
- L'image est divisée en, au moins, k régions de taille F_i. Ces régions sont disjointes et contigües, chacune d'elles sera employée pour stocker seulement un bit du message. Ce procédé garantit une distribution uniforme des bits du message dans l'image.
- Nous utilisons la clef publique du récepteur comme semence pour un générateur de nombres pseudo-aléatoire (GNPA). Ceci permet de disperser le message dans toute l'image. Le GNPA est employé pour produire des nombres pseudo-aléatoires. Ces nombres indiquent les régions qui sont employées pour insérer les bits du message.
- Dans chaque région le pixel central est pris en compte par porter l'information.
- Finalement, le LSB du pixel central de la région déterminée par le GNPA est remplacé par le bit du message.

4.2.4 Résultats

Dans cette section nous présentons les résultats de la méthode de combinaison presentée section 4.2.1. A partir de l'image originale de Lena (512×512 pixels), figure 4.16.a, nous avons appliqué notre méthode de chiffrement par flot avec une clef K de 128 bits afin d'obtenir l'image cryptée figure 4.16.b. Si nous décryptons cette image, nous pouvons noter qu'il n'y a aucune différence entre celle-ci et l'originale. Avec $K = B367EF4A5C18DA90B7E164382D90CF52$ nous obtenons les valeurs de $\alpha(\imath)$ et $p(\imath)$, montrées tableau 4.5.

Nous avons crypté la clef K de 128 bits deux fois avec l'algorithme RSA afin d'obtenir K''. Du fait de la longueur de la clef publique de B, la longueur de K'' est proche de

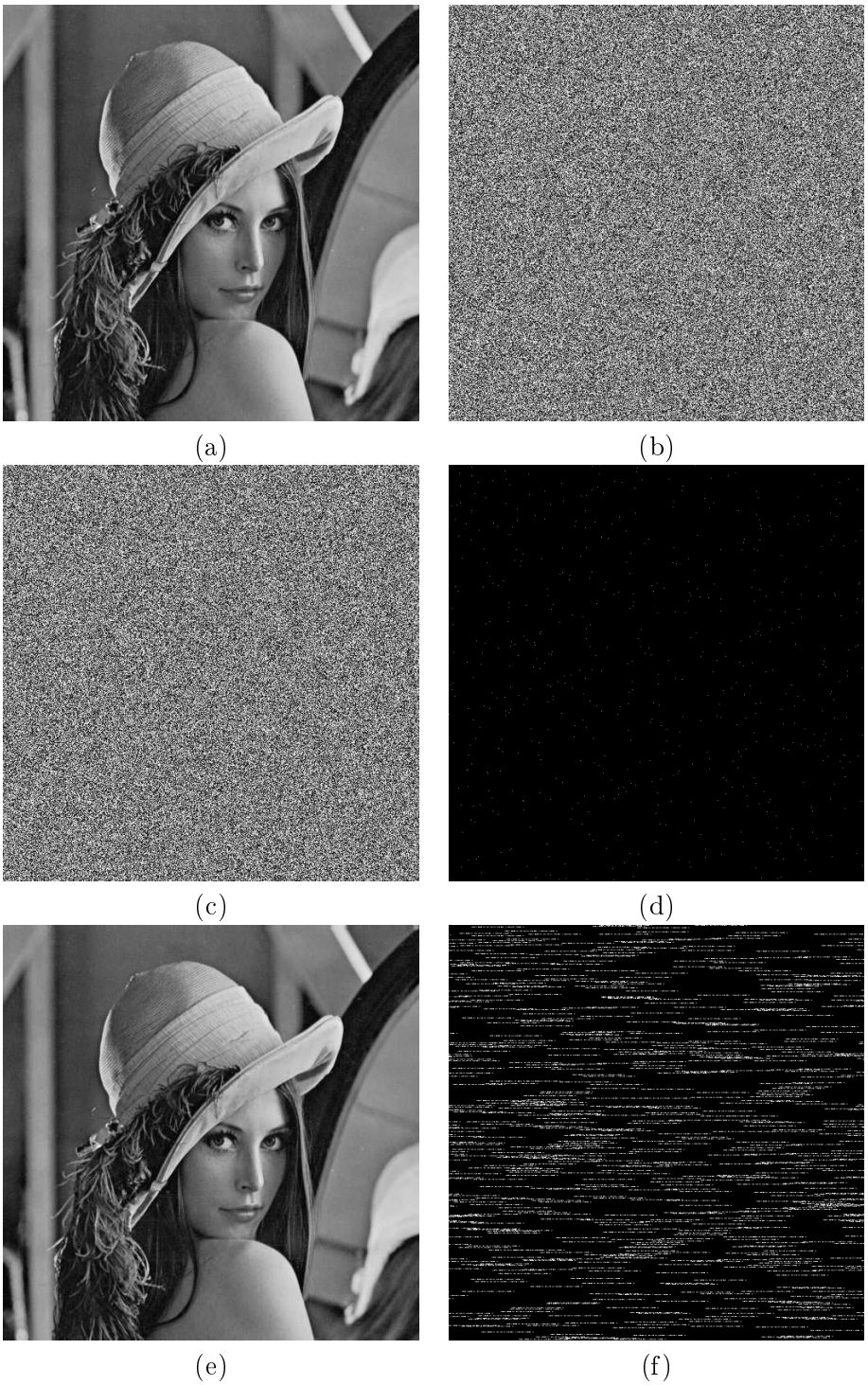


FIG. 4.16 – a) Image originale, b) Image cryptée par flot avec une clef de 128 bits, c) Image (b) marquée avec la clef secrète cryptée, d) Différence entre les images (b) et (c), e) Décryptage de l'image (c), f) Différence entre l'image originale (a) et (e).

TAB. 4.5 – Valeurs de $\alpha(\imath)$ et $p(-\imath)$.

$\alpha(\imath) =$	1 -2 -1 2 0 1 0 -2 2 1 -2 2 0 -1 1 1 0 0 -2 -1 -1 0 1 -1 2 0 1 1 1 0 -1 -1 1 -2 0 -2 2 1 -1 0 0 1 0 -1 -1 -2 1 -1 -1 1 2 0 1 0 -1 -1 2 -1 2 -2 0 0 -1 1
$p(-\imath) =$	179 102 205 155 54 108 217 179 103 207 159 63 126 253 251 247 239 222 189 122 244 233 210 165 74 148 41 82 165 75 151 46 92 184 112 224 193 131 6 12 24 49 99 198 141 27 54 109 218 181 106 212 169 82 164 72 144 33 66 133 11 22 45 91

1024 bits. Ensuite, avec la méthode d'IDC présentée section 4.2.3, nous avons inséré la clef K'' dans l'image cryptée, figure 4.16.c. Le facteur d'insertion est de 1 bit tous les 256 pixels. La différence entre l'image cryptée et l'image cryptée marquée est présentée figure 4.16.d. Les pixels utilisés pour l'IDC sont visibles, le $PSNR = 74,59\ dB$.

Finalement, après décryptage de l'image cryptée et marquée figure 4.16.c nous obtenons l'image finale illustrée figure 4.16.e. La différence entre l'image originale et l'image finale est présentée 4.16.f. Nous voyons dans cette figure que les différences entre les deux images ($PSNR = 54,72\ dB$) ont été diffusées dans toute l'image. Cependant, du fait que la valeur moyenne des coefficients $\alpha(\imath)$ est égale à zéro le bruit dû à l'IDC est atténué durant la phase de décryptage.

Afin de comparer notre résultat nous avons appliqué notre méthode d'IDC sur l'image Lena cryptée en utilisant l'algorithme AES avec le mode ECB, figure 4.17.a, et avec le mode CFB, figure 4.17.b, qui sont deux modes de chiffrement par flot. Après décryptage de l'image marquée et chiffrée par AES en mode ECB nous obtenons l'image illustrée figure 4.17.c. L'image différence entre l'image originale et l'image décryptée, figure 4.17.e, montre que les variations sont très importantes, le $PSNR = 23,27\ dB$. Nous pouvons remarquer que la largeur des blocs faux est égale à 128 bits (16 pixels). Après décryptage de l'image marquée et chiffrée par AES en mode CFB nous obtenons l'image illustrée figure 4.17.d. La qualité de l'image n'est pas bonne, $PSNR = 22,93\ dB$. La différence entre les images originale et décryptée, figure 4.17.e, montre que les variations sont diffusées dans le bloc suivant. La qualité de l'image est bonne, le $PSNR = 37,23\ dB$.

En conclusion avec notre méthode combinant cryptage et IDC il est possible d'avoir un système de transmission autonome et de garantir l'intégrité des données. Notre méthode de chiffrement asynchrone est robuste au bruit et par conséquent nous pouvons insérer un message dans une image cryptée sans perturber la phase de décryptage. Nous avons comparé notre méthode de chiffrement par flot aux modes ECB et CFB d'AES mais nous n'obtenons pas le même niveau de qualité.

4.2.5 Cryptanalyse

Dans cette section nous alons analyser la robustesse du système de chiffrement au bruit. Nous étudions l'impact du bruit inséré sur la qualité de l'image décryptée et

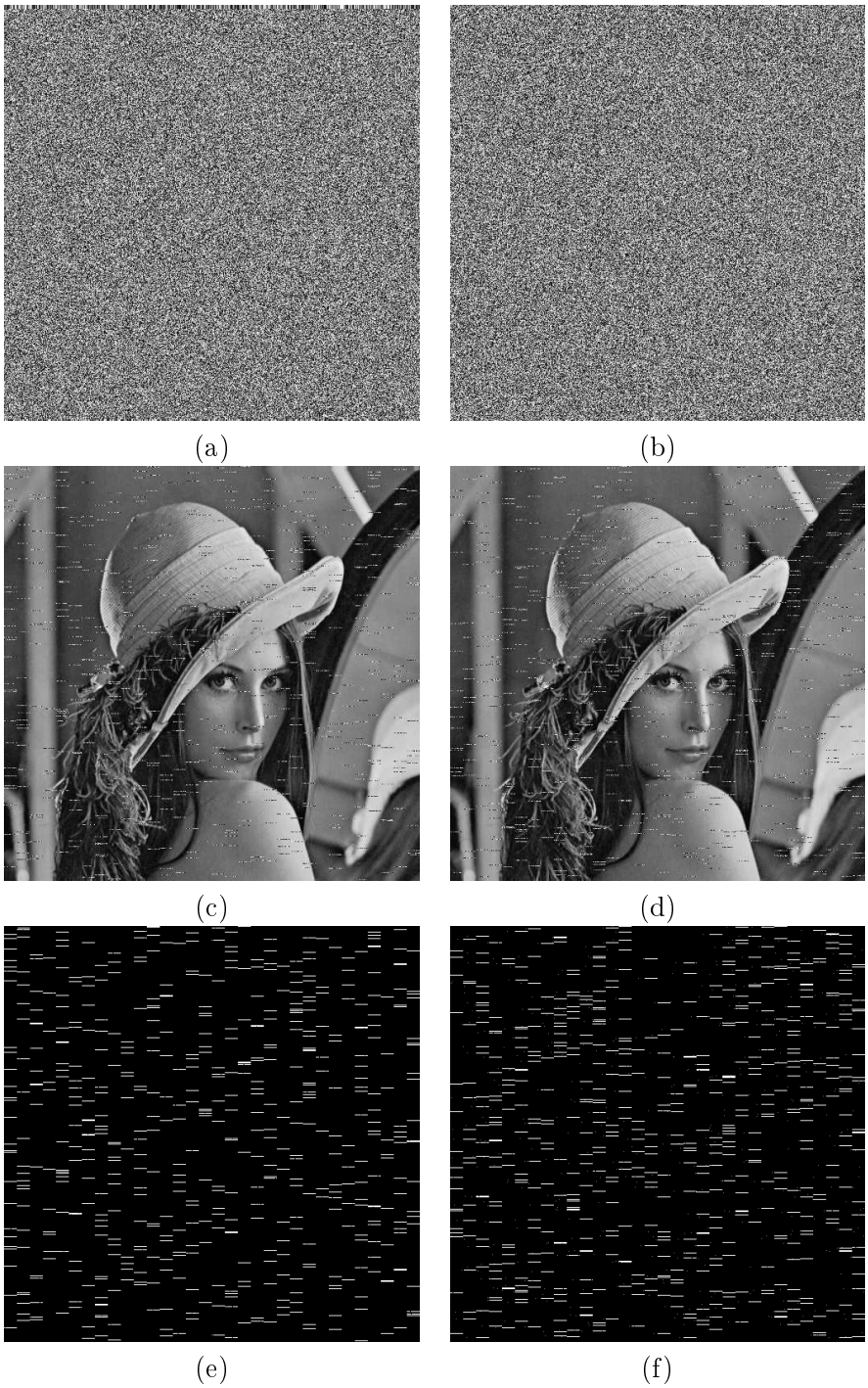


FIG. 4.17 – a) Image Lena chiffrée avec l'algorithme AES en mode ECB et marquée, b) Image Lena chiffrée avec l'algorithme AES en mode CFB et marquée, c) Résultat du décryptage de (a), d) Résultat du décryptage de (b), e) Différence entre l'image originale et (c), f) Différence entre l'image originale et (d).

l'évolution du bruit inséré dans l'image décryptée.

Dans la phase du décryptage, un pixel bruité (de l'image cryptée) se propage et modifie les $k/2$ pixels qui le succèdent dans l'image décryptée. Ceci concerne la longueur de la clef secrète utilisée dans le système de chiffrement. La fonction de décryptage représentée par l'équation 4.10 est définie pour décrypter une image cryptée sans bruit inséré. Alors que dans notre cas, nous décryptons une image cryptée bruitée, composée d'une image cryptée et d'un bruit inséré et l'augmentation du pourcentage du bruit, détériore l'image dans la phase du décryptage à cause de l'effet de l'étalement.

Donc d'après les relations 4.10 et 4.11, la fonction du décryptage en fonction du bruit devient :

$$\tilde{p}_i = (\tilde{p}'_i - \tilde{z}_i) \mod 256. \tag{4.17}$$

D'après l'équation 4.10, on peut écrire :

$$\tilde{z}_i = (\sum_{j=1}^{k/2} \alpha_j \tilde{p}'_{i-j}) \mod 256 \tag{4.18}$$

$$= (\sum_{j=1}^{k/2} \alpha_j (p'_{i-j} + n_{i-j})) \mod 256$$

$$= (\sum_{j=1}^{k/2} \alpha_j p'_{i-j} + \sum_{j=1}^{k/2} \alpha_j n_{i-j}) \mod 256$$

Dans ce résultat, on retrouve l'équation 4.10, donc :

$$\tilde{z}_i = (z_i + \sum_{j=1}^{k/2} \alpha_j n_{i-j}) \mod 256$$

On pose le flux de clefs du bruit comme :.

$$Z_i = \sum_{j=1}^{k/2} \alpha_j n_{i-j}$$

On déduit alors que :

$$\tilde{z}_i = (z_i + Z_i) \mod 256.$$

D'après l'équation 4.17 et l'équation précédente, on a :

$$\tilde{p}_i = (\tilde{p}'_i - z_i - Z_i) \mod 256$$

$$= (p'_i + n_i - z_i - Z_i) \mod 256$$

$$= (z_\imath + p_\imath + n_\imath - z_\imath - Z_\imath) \mod 256$$

Après simplification, on retrouve que :

$$\tilde{p}_\imath = (p_\imath + n_\imath - Z_\imath) \mod 256$$

Enfin, d'après l'équivalence avec relation $\tilde{p}_\imath = (p_\imath + \tilde{n}_\imath)$, on déduit que :

$$\tilde{n}_\imath = n_\imath - Z_\imath \tag{4.19}$$

D'après l'équation 4.19 le nouveau bruit dans l'image décryptée bruitée ne dépend pas de pixels cryptés. Il dépend essentiellement du bruit initial (inséré dans l'image cryptée bruitée) et du flux de clefs du bruit qui présente un ordre de récurrence correspondant à la longueur $k/2$ de la clef K. Ce qui explique le phénomène de propagation du nouveau bruit sur les $k/2$ pixels qui le succèdent. Cette information est très importante pour les cryptanalistes puisqu'une analyse du bruit fournit l'information sur la taille de la clef de chiffrement.

4.2.6 Bilan

Dans cette section, nous avons présenté une méthode qui combine cryptage et IDC. Il est ainsi possible d'avoir un système de transmission autonome permettant de garantir l'intégrité des données. Nous avons utilisé à la fois les avantages du cryptage symétrique et asymétrique. Dans l'approche autonome proposée, nous avons choisi de chiffrer avec notre méthode de chiffrement par flot asynchrone et de chiffrer la clef secrète avec un algorithme asymétrique.

Nous avons analysé la robustesse au bruit de notre méthode de cryptage appliquée sur une image médicale. Nous constatons que notre méthode est robuste à une certaine quantité de bruit. Pour insérer la clef secrète cryptée nous avons utilisé une méthode d'IDC dans le domaine spatial qui remplace le LSB. Nous remarquons que quelque soit le type d'image (avec ou sans zones homogènes), aucune texture n'apparaît dans les images cryptées.

Finalement nous avons présenté un résultat complet de la méthode et avons comparé dans la combinaison notre algorithme de chiffrement aux modes de chiffrement par flot de l'algorithme AES. Notre méthode de chiffrement permet de conserver une meilleure qualité pour l'image finale. De plus, les calculs qui composent notre méthode sont peu nombreux, notre méthode s'avère donc très rapide.

Nous avons vu dans le chapitre précédent que les attaques statistiques n'étaient pas possible du fait d'une entropie élevée. Mais en perspective de cette méthode de combinaison nous pensons évaluer d'autres types d'attaques afin de garantir un meilleur niveau de sécurité. Une autre solution, pour laquelle nous avons opté, consiste à utiliser une méthode de l'IDC additive pour l'insertion du message.

Conclusion

Dans ce chapitre nous avons présenté deux nouvelles méthodes de codage hybride. La première méthode combine IDC, cryptage et compression pour la protection et l'authentification des images médicales et des données des patients. Une fois le compromis espace pour l'IDC et compression évalué la méthode offrant les résultats en adéquation peut être retenue.

La deuxième méthode a traité du transfert autonome d'images en utilisant le chiffrement symétrique et asymétrique. Cette méthode a fait l'objet d'un brevet [PR05c] Il existe de nombreuses techniques développées séparément dans ces trois domaines du cryptage, de l'IDC et de la compression d'images. Cependant, notre travail s'est orienté dans le développement de nouvelles méthodes qui combinent ces trois domaines. Des cryptanalyses de cet algorithme ont été effectuées dans le cadre des travaux de master [Dje06]. Cette cryptanalyse permet à partir d'une autocorrélation entre l'image originale et l'image finale bruitée et décrypté de mettre en évidence que le bruit de l'image finale n'est plus un bruit blanc gaussien. En fait une transformée de Fourier de la fonction d'autocorélation permet d'obtenir l'information de la longueur de la clef. Cette information (longueur de clef) est un élément pertinent pour la cryptanalyse.

Chapitre 5

Cryptage Sélectif

Sommaire :

5.1	Codage entropique du JPEG
5.2	Chiffrement AES en mode CFB
5.3	Travaux précédents de CS par DCT
5.4	Approche proposée
5.5	Résultat
5.5.1	-Transfert sécurisé d'images médicales
5.5.2	-Protection de BDD de peintures numériques
5.5.3	-Protection de visages dans des séquences d'images

Introduction

Classiquement il y a deux approches pour crypter des données. La première est un cryptage complet où toutes les informations sont chiffrées. L'inconvénient de ce type de cryptage est qu'il est toujours appliqué de la même manière, quel que soit l'application et le niveau de sécurité souhaité. La seconde approche de cryptage est adaptée au niveau de protection désiré. Le chiffrement est adapté à la sécurité désirée afin de réduire les ressources informatique et le temps de calcul disponible. C'est dans cette seconde approche que nous trouvons le cryptage sélectif. Le cryptage sélectif (CS) est une approche qui ne chiffre qu'une partie des données afin de diminuer le temps de calcul. Les utilisateurs peuvent appliquer une sécurité proportionnelle ou réglable en fonction du niveau de protection désiré [NPP+03].

Un nombre important d'applications peut se contenter d'un niveau inférieur à un cryptage complet en utilisant un CS. Nous pouvons citer de nombreuses applications ou par exemple des parties de l'image doivent être visibles pour autoriser une recherche et une classification de données. Des applications dans le domaine de la formation présentent des images qui doivent être partiellement visibles sans révéler complètement toute l'information. Les peintures numériques doivent être présentées sur Internet avec une qualité visible réglable. Le transfert de photos depuis des téléphones portables peut également se contenter d'un cryptage partiel afin d'assurer la confidentialité. C'est aussi le cas des images médicales prises depuis un appareil médical et devant être envoyées sur le réseau afin d'établir un diagnostic à distance. De plus, l'appareil d'acquisition des images médicales peut se trouver dans une ambulance ou dans tout autre véhicule mobile, et dans ce cas la transmission est effectuée par l'intermédiaire de réseaux sans fil. Pour des raisons vitales, dans ce type d'applications, les images doivent être transmises rapidement et surement, et dans ce cas un CS semble être la meilleure solution (compromis temps/sécurité).

Ce chapitre présente une nouvelle approche de CS basée sur les travaux de Droogenbroeck et Benedett [DB02] pour des images comprimées au format JPEG. Notre approche est basée sur le cryptage par AES de certains flux binaires issus du codage entropique plus précisément du vecteur d'Huffman. Nous montrons l'application de notre méthode sur des images médicales [RPDM06], une BBD (base de données) de peintures numériques [RPB06a] et la protection des visages dans des séquences d'images [RPM+06, RPB06b].

5.1 Codage entropique du JPEG

Le format standard JPEG a été présenté section 3.5.1.1. Après la quantification, les coefficients DCT sont lus dans un ordre prédéfini en zigzag en partant des basses fréquences et en terminant par les plus hautes fréquences, figure 5.1. L'approche proposée est entièrement réalisée dans cette étape de codage. Nous allons donc détailler le codage d'Huffman et donner un exemple pratique de son fonctionnement.

Dans le codage d'Huffman les coefficients quantifiés sont codés par des couples {(HEAD),(AMPLITUDE)}. L'entête HEAD contient des contrôleurs obtenus par les

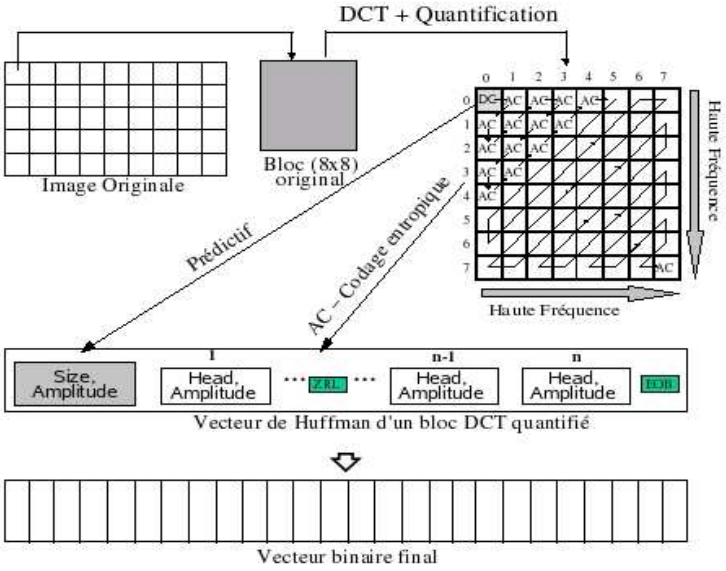


FIG. 5.1 – L'algorithme JPEG.

tableaux d'Huffman pour la compression et la décompression. Le paramètre AMPLITUDE est un entier signé correspondant à l'amplitude d'un coefficient AC non nul. La structure HEAD varie en fonction du type de coefficient. Pour les AC il est composé de (RUNLENGTH, SIZE), alors que pour les DC il est composé seulement de la taille SIZE.

Coefficient DC

Pour le coefficient DC l'amplitude est codée par la différence entre deux coefficients voisins.

Les coefficients DC transportent une information visible importante et une corrélation locale significative. Ils sont hautement prédictibles, ainsi JPEG traite les coefficients DC séparément des 63 coefficients AC. La valeur des composantes DC est importante (rarement nulle) et variée, mais est souvent très proche de celle de ses voisins. La seule valeur qui est donc encodée est la différence $DIFF$ entre le coefficient DC_i quantifié du bloc courant i et le précédent DC_{i-1}. Les blocs sont lus de la gauche vers la droite, ligne par ligne :

$$DIFF = DC_i - DC_{i-1}.$$

L'approche proposée est basée essentiellement sur le cryptage de certains coefficients AC. Pour cela, la description du codage d'Huffman des coefficients AC est plus détaillée.

Coefficients AC

JPEG utilise une méthode alternative intelligente de codage des AC, basée sur la combinaison des informations longueur des séquences et amplitude. Cette approche agrège les coefficients nuls quand il y a des plages de zéros. La valeur RUNLENGTH correspond au nombre de coefficients AC qui ont pour valeur zéro précédant une valeur non nulle dans la séquence en zigzag. La taille SIZE est la quantité de bits nécessaires pour représenter la valeur de l'amplitude.

Deux codes particuliers correspondant à (RUNLENGTH, SIZE) = $(0, 0)$ et $(15, 0)$ sont utilisés pour symboliser respectivement la fin d'un bloc (EOB) et la longueur d'une plage de zéros (ZRL). Le symbole EOB est transmis après le dernier coefficient non nul du bloc quantifié. C'est ainsi le chemin le plus efficace pour coder la fin d'une plage de zéros. Ceci peut être vu comme un symbole de sortie qui termine le bloc 8×8. Dans le processus de décodage, quand un symbole EOB est trouvé, tous les coefficients restants du bloc sont initialisés à zéro. Le symbole EOB est omis dans le cas ou l'élément final du vecteur est non nul. Le symbole ZRL est transmis quand la valeur du RUNLENGTH est plus grande que 15 et il représente une longueur de plage de 16 zéros.

Nous présentons un exemple pratique, illustré tableau 5.1, qui montre un bloc original 8×8 de coefficients DCT quantifiés. Le tableau 5.2 présente le résultat du codage d'Huffman. Si nous suivons le parcours en zigzag dans le bloc présenté tableau 5.1, le premier coefficient AC non nul est -5 sans valeur à zéro le précédant. Ceci produit une représentation intermédiaire de $(0, 3)(-5)$, où 3 est le nombre de bits nécessaire pour coder $| -5 |$. Ensuite le coefficient AC suivant est 8, qui n'est pas non plus précédé de zéro. Par conséquent sa représentation intermédiaire est $(0, 4)(8)$. Les coefficients AC suivant sont deux zéros consécutifs suivis de la valeur 2, donc le (RUNLENGTH,SIZE)(AMPLITUDE) est $(2, 2)(2)$. La valeur du AC suivant à coder est -1 précédée de 3 zéros, donc sa représentation intermédiaire est $(3, 1)(-1)$. Après nous avons $RunLength > 15$, donc nous devons utiliser le symbole ZRL représentant 16 zéros successifs. Le dernier coefficient non nul est 1 précédé par un zéro, donc $(1, 1)(1)$. Comme c'est le dernier coefficient non nul, le symbole final de ce bloc 8×8 est la marque EOB.

97	-5	2	0	0	0	1	0
8	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

TAB. 5.1 – Un bloc de coefficients DCT quantifiés.

Après avoir construit la représentation intermédiaire d'Huffman, il est nécessaire de trouver le *Mot de code* pour la valeur de l'entête HEAD dans le tableau 5.3.a et de

Représentation de Huffman intermédiaire		Flux binaire d'Huffman	
HEAD	AMPLITUDE	HEAD	AMPLITUDE
(0,3)	-5	100	010
(0,4)	8	1011	1000
(2,2)	2	11111001	10
(3,1)	-1	111010	0
(ZRL)	—	11111111001	—
(1,1)	1	1100	1
(EOB)	—	1010	—

TAB. 5.2 – Le flux binaire comprimé pour les coefficients AC du tableau 5.1.

Run/Size	Longueur des codes	Mots de code
0/0 (EOB)	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
⋮	⋮	⋮
1/1	4	1100
1/2	5	11011
1/3	7	1111001
⋮	⋮	⋮
2/1	5	11100
2/2	8	11111001
⋮	⋮	⋮

Mots de code pour les coefficients AC
(a)

Taille (k) en bits	Intervalles (-y..-x)	(x..y)
1	-1	1
2	-3,-2	2,3
3	-7...-4	4...7
4	-15...-8	8...15
5	-31...-16	16...31
6	-63...-32	32...63
7	-127..-64	64...127
8	-255...-128	128...255
9	-511...-256	256...511
10	-1023...-512	512...1023

Taille pour les coefficients AC
(b)

TAB. 5.3 – Tableaux du codage entropique du JPEG.

calculer la représentation finale de la valeur AMPLITUDE en utilisant le tableau 5.3.b. Par conséquent, le mot de code pour $(0,3)$ pris du tableau 5.3.a est la séquence de bits 100 qui est appelée HEAD. Pour $(0,4)$ la séquence 1011 est utilisée, et ainsi de suite comme montré dans le tableau 5.2. La représentation finale de l'AMPLITUDE est calculée de la manière suivante. Le bit le plus à gauche est toujours utilisé pour le signe. Le bit de signe est égal à 0 pour les valeurs négatives et à 1 pour les valeurs positives. Premièrement nous cherchons l'intervalle dans le tableau 5.3.b, pour (-5) par exemple, nous avons la taille en bits $k = 3$. Cela signifie que nous avons un bit pour le signe et deux pour représenter la valeur. Cette représentation est calculée par l'équation $V = |AMPLITUDE| - 2^{k-1}$. Si la valeur est négative le codage d'Huffman utilise la notation avec le complément à 1 telle que tous les bits changent de valeurs. L'expression booléenne est $V = NOT(V)$. Pour -5 par exemple, nous avons $V = 5 - 2^2 = 1$, qui en binaire (sur 2 bits) est 01. La valeur -5 est négative donc le bit de signe est égal à 0, $V = NOT(01)$ et $V = 10$. La représentation finale en binaire de -5 est 010. Pour 8, nous avons $k = 4$ et par conséquence $V = 8 - 2^3 = 0$ et comme 8 est positif la représentation binaire finale est 1000 comme montré dans le tableau 5.2.

Donc, pour chaque bloc 8×8, la séquence zigzag des 63 coefficients AC quantifiés est une séquence de bits qui peut être des paires de (HEAD, AMPLITUDE) ou des marques spéciales, EOB or ZRL. Avec l'exemple donné le résultat du flux binaire d'Huffman est : 100 010 1011 1000 11111001 10 111010 0 11111111001...

5.2 Chiffrement AES en mode CFB

L'algorithme standard de chiffrement AES a été présenté section 1.4.2. Il peut supporter les modes de chiffrement ECB, CBC, OFB, CFB et CTR. Les blocs de données et les clefs peuvent être de longueur de 128, 192 ou 256 bits. Dans notre approche nous utilisons des clefs et des blocs de données de 128 bits et le mode CFB. Dans ce mode, le flux de clefs z_i est obtenu en cryptant le bloc chiffré précédent y_{i-1}. La première ronde est déclenchée avec le vecteur d'initialisation VI. Soient le cryptage $Y_0 = VI$, $z_i = e_K(Y_{i-1})$, $Y_i = z_i \oplus X_i$ et le décryptage $z_i = e_K(Y_{i-1})$, $X_i = z_i \oplus Y_i$, comme montré figure 5.2.

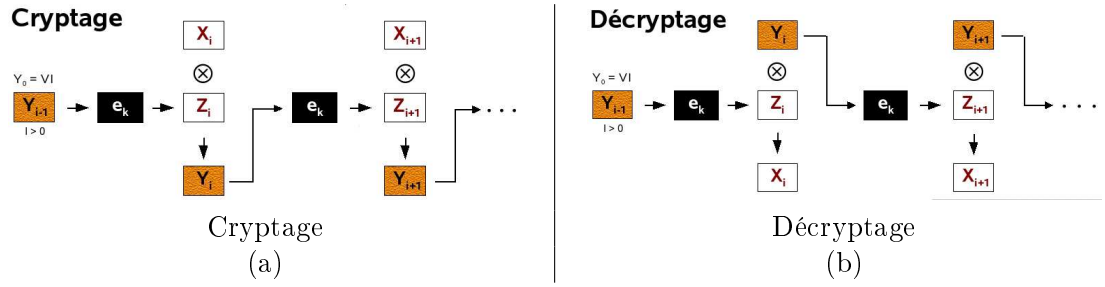


FIG. 5.2 – Le mode CFB de l'algorithme AES.

Il est important de noter que la fonction de cryptage e_K est utilisée pour la phase de

cryptage (figure 5.2.a) mais également pour la phase de décryptage (figure 5.2.b) dans le mode CFB.

5.3 Travaux précédents de CS par DCT

Le format JPEG est encore le format le plus utilisé pour la compression d'images. Actuellement, il a été développé sur des quantités de cartes dédiées à la compression pour les caméras numériques, les téléphones portables, les scanners, les machines mobiles. De nombreuses méthodes de CS ont été créées avec une approche de cryptage pour des images codées par transformée en cosinus discrète (DCT).

- Tang [Tan96] a proposé une technique appelée permutation zigzag applicable à des vidéos ou des images basées DCT. Bien que sa méthode offre plus de confidentialité, elle diminue le taux de compression.
- Fisch *et al.* [FSU04] ont proposé une méthode telle que les données sont organisées dans une forme de flux binaire réglable. Ces flux binaires sont construits avec les coefficients DC et quelques coefficients AC de chaque bloc et sont arrangés dans des couches en fonction de leur importance visuelle. Le cryptage partiel est alors effectué au niveau de ces couches.
- D'autres méthodes ont été développées spécifiquement pour les vidéos [AARAS99, ZL99, CL00, WSZ+02].
- Récemment, Said a montré la force des méthodes de cryptage partiel en testant des attaques qui exploitent l'information non cryptée de l'image associée à une image de petite taille [Sai05].
- Notre approche est basée sur ce travail de Droogenbroeck et Benedett [DB02]. Ils sont à l'origine d'une technique qui crypte les coefficients ACs. Dans leur méthode, les coefficients DCs ne sont pas cryptés car ils portent une information visible importante mais sont hautement prédictibles. De plus, le taux de compression est constant et conserve le format du flux binaire JPEG. Par contre leur méthode possède les désavantages suivants :

  1. la compression et le cryptage sont fait séparément et par conséquent leur méthode prend plus de temps que la compression seule ;
  2. tous les coefficients AC de l'image sont cryptés, il n'existe pas une approche de réglable pour le cryptage ou décryptage ;
  3. le mode de cryptage utilisé est le mode par bloc, c'est-à-dire que plusieurs blocs (8×8) sont utilisés comme entrée pour l'algorithme de chiffrement, par conséquence le cryptage ou décryptage d'une seule région d'intérêt (ROI) est difficile ;
  4. les blocs identiques sont chiffrés d'une façon identique.

Dans la section suivante nous présentons une nouvelle approche du travail [DB02]. Notre approche résout les problèmes cités ci-dessus en engageant un mécanisme d'amélioration.

5.4 L'approche proposée

L'idée principale de la méthode proposée est illustrée figure 5.3 et composée de 3 étapes :

1. Prendre les valeurs des AMPLITUDES des coefficients AC non nuls du flux binaire d'Huffman, des plus hautes fréquences vers les basses fréquences (ordre zigzag inverse) afin de construire le vecteur du message en clair X.
2. Coder X avec l'algorithme AES en mode CFB.
3. Substituer le flux binaire d'Huffman par l'information équivalente cryptée qui est de même taille.

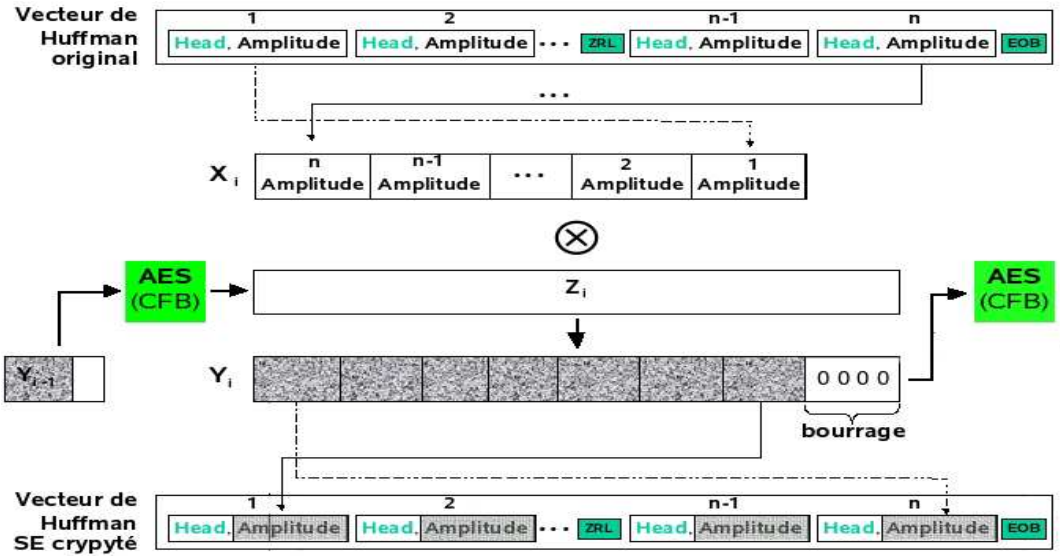


FIG. 5.3 – Présentation générale de la méthode proposée.

Ces opérations sont appliquées séparément pour chaque bloc DCT quantifié.

5.4.1 Quelques considérations

Avant de présenter en détail la méthode, nous souhaitons prendre en compte quelques considérations.

Marques de contrôle

Le vecteur d'Huffman est composé de couples {HEAD, AMPLITUDE} et de marques de contrôle ZRL et EOB. Comme nous avons vu, ces marques de contrôle n'apparaissent pas obligatoirement. Un bloc avec tous les coefficients à zéro n'a pas de {HEAD, AMPLITUDE}. Un bloc avec tous les coefficients différents de zéro n'a pas ZRL et EOB. Dans notre méthode de cryptage sélectif, nous ne modifions rien dans les parties HEAD

ainsi qu'au niveau des marques de contrôles indiquées. Pour garantir une compatibilité totale avec tous les décodeurs, le flux binaire doit seulement être modifié dans les zones ou cela ne compromet pas les souhaits du format original JPEG.

Ordre zigzag inverse

La raison de construire un chemin des hautes fréquences vers les basses fréquences (ordre zigzag inverse) vient du fait que les caractéristiques visuelles les plus importantes de l'image se situent dans les basses fréquences, alors que les détails sont localisés dans les hautes fréquences. Le système visuel humain (SVH) est plus sensible aux basses fréquences qu'aux hautes fréquences. Cependant, nous pensons qu'il est intéressant de pouvoir calibrer l'apparence visuelle de l'image résultante. Cela signifie que nous nous orientons vers une méthode de cryptage réglable qui peut augmenter jusqu'à se rapprocher fortement de la composante DC de chaque bloc (basses fréquences).

Contrainte C

Une caractéristique concernant la quantité maximale de bits utilisés pour construire le texte clair X est à prendre en compte. Cette caractéristique règle le niveau de cryptage et la qualité visuelle de l'image résultat. Si rien n'est stipulée, la valeur du nombre de bits chiffrés est la taille du bloc chiffré $n = 128$. La taille du bloc est une contrainte dans le sens que nous ne pourrons pas chiffrer plus de $n = 128$ bits par bloc. La contrainte minimale est 8. Tous les blocs qui possèdent moins de 8 bits à crypter ne sont pas utilisés dans notre méthode.

Bourrage

En codage, le bourrage (padding) est une méthode permettant d'additionner des textes clairs de longueur variable. Ceci est nécessaire car le cryptage travaille sur une taille binaire fixée (128 bits dans notre méthode), mais la longueur du message crypté peut varier. Dans le mode CFB le bloc précédemment crypté est utilisé comme entrée pour l'algorithme AES. Dans notre méthode les blocs sont variables. Nous appliquons donc la fonction de remplissage (padding) $p(j) = 0, \forall j \in \mathbb{Z}$ tel que $\{\varphi < j \leq 128\}$, afin de remplir si nécessaire avec des zéros le vecteur Y_i.

Certains systèmes complexes de bourrage existent mais nous utiliserons le plus simple, en rajoutant des bits à zéros afin d'atteindre la longueur de bloc souhaitée. Historiquement, le bourrage est utilisé afin de rendre la cryptanalyse plus difficile, mais actuellement le bourrage est plus utilisé pour des raisons techniques avec les chiffrements par bloc.

Choix du mode CFB

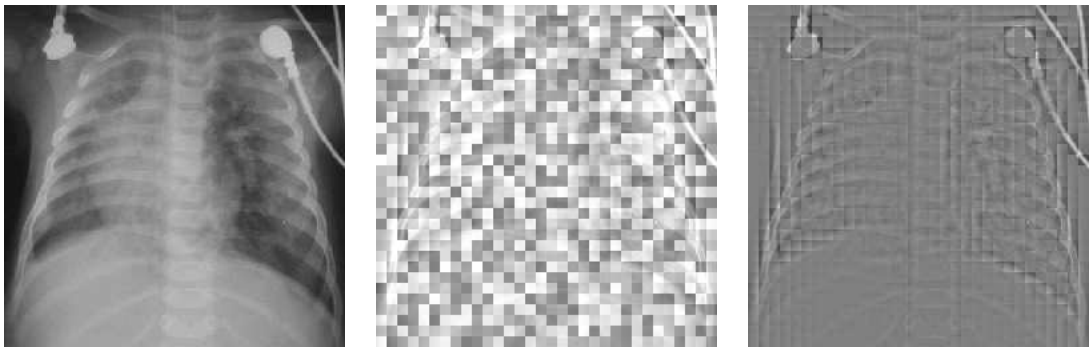
Le choix du mode CFB est basé sur l'efficacité du décryptage. Même si AES est un algorithme de chiffrement par bloc, les modes OFB, CFB et CTR opèrent comme des chiffrements par flot. Ces modes ne nécessitent aucune mesure particulière concernant

la longueur des messages qui ne correspond pas à une longueur multiple de la taille d'un bloc puisqu'ils travaillent tous en effectuant un ou exclusif entre le texte clair et la sortie du chiffrement par bloc. L'utilisation des modes de chiffrement par bloc (ECB et CBC) nous ne convient donc pas parce que dans notre approche les blocs sont variables et dépendent du contenu de l'image.

Les modes par flot OFB et CTR sont aussi applicables dans notre méthode, cependant pour décrypter le bloc Y_i dans ces modes, nous avons besoin de reconstruire toutes les sous-clefs à partir du vecteur d'initialisation VI jusqu'à i. Ceci est dû au fait que le flux de clef z_i est obtenu en cryptant le précédent flux de clef $z_0 = VI$, $z_i = e_K(z_{i-1})$. Contrairement aux autres modes, dans le mode CFB pour décrypter le bloc Y_i, il est nécessaire d'avoir seulement le bloc précédemment crypté Y_{i-1} et la clef secrète K.

Non utilisation du coefficient DC

Les coefficients DC transportent une information visible importante et ils sont hautement prédictibles. La figure 5.4 montre l'utilisation des coefficients DC pour la dissimulation d'information. Dans l'image originale figure 5.4.a seuls les coefficients DC ont été cryptés. Nous pouvons noter que visiblement ce cryptage partiel semble efficace. Cependant, il est très facile de découvrir le contenu de l'image [PR05b, PR05a].



(a) (b) (c)

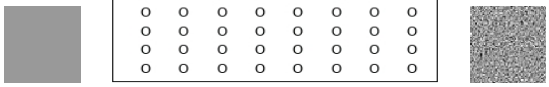
FIG. 5.4 – a) Image originale, b) Cryptage des coefficients DC, c) Remplacement des DCs par zéro.

Nous avons fait quelques essais pour deviner les valeurs des DC dans l'image chiffrée. Par exemple, si nous remplaçons les valeurs des DCs de chaque bloc chiffré par zéro, nous obtenons l'image illustrée figure 5.4.c. Nous constatons qu'il est alors possible de devenir le contenue de l'image médicale. Par l'intermédiaire de cette attaque, il est donc possible de faire ressortir tous les détails de l'image qui sont en clair. Une seconde étape de cryptanalyse consiste simplement à considérer que 2 valeurs voisines de DC sont proches (attaque du puzzle) afin de reconstruire l'image originale.

Nous pouvons conclure que, si seulement la composante DC a été chiffrée, il est très facile d'accéder à l'information visuelle de l'image de JPEG en remplaçant simplement les coefficients DC chiffrés par une valeur constante.

Homogénéité

Plus un bloc de l'image originale est homogène, plus il y a des zéros au niveau des coefficients AC quantifiés et plus les coefficients AC ont des petites amplitudes. En effet, la DCT (Discrete Cosine Transform) sépare l'image en sous-bandes spectrales. Donc les régions de l'image qui sont monotones fourniront des coefficients DCT proches de zéro qui après la quantification deviendront nuls. La figure 5.5 présente les coefficients après la DCT et la quantification pour un bloc homogène figure 5.5.a et pour un bloc texturé figure 5.5.c.



72	1	0	0	0	0	0	0
-2	-1	0	0	0	0	0	0
3	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

197	13	81	-68	-25	22	-21	18
25	31	22	-42	33	35	-30	5
24	-41	28	-55	-66	21	8	-52
12	32	63	-19	-108	31	68	-86
42	-16	-19	-37	-5	36	-7	-56
-30	33	79	17	23	-16	-10	86
-2	52	24	-35	5	15	-14	10
14	-25	-2	22	-12	6	21	-26

(a) (b) (c) (d)

FIG. 5.5 – a) Bloc homogène, b) Coefficients DCT quantifié de (a), c) Bloc texturé, d) Coefficients DCT quantifié de (c).

Le bloc homogène génère des coefficients (figure 5.5.b.) avec des petites amplitudes et beaucoup de valeurs à zéros. Dans notre approche, si le nombre de bits dans un bloc est inférieur à 8, alors le bloc n'est pas utilisé. Ceci rend efficace le cryptage pour les images avec une quantité importante de régions homogènes. Notons que dans un bloc homogène, il est très difficile de cacher l'information du fait de l'information quasi identique contenue dans les blocs voisins. Par conséquent, dans le cas d'un CS nous ne prendrons pas en compte les blocs homogènes pour le chiffrement.

Le bloc texturé produit des coefficients avec des amplitudes importantes (figure 5.5.d.). Ce sont les blocs les plus cryptés. Les blocs texturés (où contenant des contours) détiennent donc un rôle important pour la qualité visuelle de l'image finale cryptée partiellement.

5.4.2 Cryptage sélectif

Le procédé de cryptage sélectif est donc composé de trois étapes : la **construction du texte clair** X_i, la procédure de **cryptage par AES** pour créer Y_i et la **substitution du flux binaire d'Huffman** par l'information cryptée.

5.4.2.1 Construction du texte clair X

Pour construire le texte clair X_i, nous prenons les coefficients AC non nuls du bloc courant i en accédant au vecteur d'Huffman de la fin vers le début afin de créer des paires {HEAD, AMPLITUDE}. De chaque entête HEAD nous obtenons la longueur de l'AMPLITUDE en bit. Ces valeurs sont calculées à partir de l'équation (5.1). Comme

montré dans la vue générale de la méthode proposée figure 5.3, seule les AMPLITUDEs $(A_n, A_{n-1}...A_1)$ sont prises en compte pour construire le vecteur X_i. La longueur en bits du texte clair final L_{X_i} dépend, à la fois, de l'homogénéité φ du bloc et de la contrainte donnée C. Cette contrainte C spécifie la quantité maximale de bits qui doit être prise en compte dans chaque bloc. D'un autre côté, l'homogénéité dépend du contenu de l'image et spécifie la quantité de données en bits de chaque bloc. Cela signifie qu'un bloc avec un grand φ va produire un grand L_{X_i}. Le vecteur d'Huffman est traité tant que $L_{X_i} \leq C$ et que le coefficient DC n'est pas atteint. Si $\varphi < 8$ le bloc n'est pas crypté.

$$\begin{cases} L_{X_i} = C & Si \quad \varphi \geqslant C \\ L_{X_i} = \varphi & Si \quad 8 \leq \varphi < C \\ \\ \text{et} & C \in \{128,\ 64,\ 32,\ 16,\ 8\} \text{ bits.} \end{cases} \tag{5.1}$$

5.4.2.2 Cryptage par AES

Le cryptage de la méthode proposée est appliqué en même temps que le processus de codage entropique durant la création du vecteur d'Huffman. Cependant, notre méthode peut être appliquée sur tous les systèmes de codage JPEG utilisant la table d'Huffman, décrite précédemment.

Construction du vecteur d'initialisation VI

Le vecteur VI, pour la première itération, est créé à partir de la clef secrète K avec la stratégie suivante : la clef secrète K est utilisée comme une semence pour un générateur de nombres pseudo-aléatoire (GNPA). Cette clef K est divisée en 16 portions de 8 bits chacune. Le GNPA produit 16 nombres aléatoires qui définissent l'ordre de formation du vecteur VI. Par exemple si le premier nombre aléatoire généré est 7, le premier octet de la clef secrète sera copié dans le septième élément du vecteur VI. Si le second nombre aléatoire généré est 10, le second octet de la clef occupera le $10^{ème}$ octet dans le VI et ainsi de suite. Après avoir généré le vecteur $Y_0 = VI$, il est crypté avec AES pour généré Z_0 et démarrer la procédure de cryptage. Ensuite chaque Z_i est additionné par un ou exclusif avec le texte en clair X_i pour générer Y_i, tel que le montre le schéma de chiffrement CFB, figure 5.2.a.

Dans notre méthode, nous utilisons comme entrée pour AES une clef secrète et un bloc de données avec une taille de 128 bits. Dans l'étape de chiffrement dans le mode CFB, le texte précédemment crypté est utilisé pour générer Z_i. Comme la longueur de ce bloc précédemment crypté est variable, il peut être soumis a une opération de bourrage présentée figure 5.3, si $L_{Y_{i-1}} < 128$.

5.4.2.3 Substitution du flux binaire d'Huffman

L'étape finale est la substitution de l'information initiale par l'information chiffrée dans le vecteur d'Huffman. Comme dans la première étape (construction du texte clair X_i), le vecteur d'Huffman est lu depuis la fin vers le début, mais le vecteur chiffré Y_i

est lu du début vers la fin. Connaissant la longueur en bits de chaque AMPLITUDE $(A_n, A_{n-1}...A_1)$, nous commençons par couper ces portions dans Y_i pour remplacer les valeurs AMPLITUDE dans le vecteur d'Huffman. La quantité totale de bits doit être L_{X_i}.

5.4.3 Procédure de décryptage

La procédure de décryptage est considérée comme le processus inverse du cryptage sélectif. Pour le récepteur qui possède l'image cryptée sélectivement et la clef secrète il peut choisir le décryptage total ou un décryptage partiel.

Décryptage total

Le décryptage total est le procédé pour obtenir le bloc original totalement décrypté. La clef secrète est utilisée pour construire le vecteur d'initialisation VI selon l'approche exposée section 5.4.2.2. Après avoir créé le vecteur VI, il est crypté avec AES pour générer Z_0. Dans le mode CFB, la fonction utilisée pour le décryptage et le cryptage est la même. Ensuite Z_0 est additionné par un ou exclusif avec le Y_0 pour générer le texte en clair X_0.

Pour les autres blocs $i > 0$, il suffit de prendre le bloc précédemment crypté Y_{i-1} comme entrée pour l'algorithme AES. Comme la longueur du bloc précédemment crypté est variable, il est nécessaire de faire une opération de bourrage si $L_{Y_{i-1}} < 128$. La valeur 128 est la longueur standard du bloc d'entrée pour notre méthode. Après l'obtention du X_i, il suffit de substituer l'information en clair par l'information chiffrée. Le vecteur en clair X_i est lu depuis la fin vers le début et le vecteur d'Huffman est lu du début vers la fin. Ceci rend la construction visuelle de l'image depuis la basse vers la plus haute qualité.

Décryptage partiel

Le décryptage partiel est le déchiffrement réglable. Nous pouvons choisir le niveau de décryptage partiel D_p, par rapport la contrainte de cryptage $\{128, 64, 32, 16, 8\}$, tel que $D_p < C$.

Le procédé de décryptage partiel suit les mêmes étapes du décryptage total. La différence est la quantité d'information en clair qui sera substituée dans le vecteur d'Huffman. Pour une contrainte de cryptage $C = 128$ et décryptage partiel de $D_p = 32$ par exemple, 1/4 des bits de X_i sont pris en compte pour la substitution dans le vecteur de Huffman. Le parcours est le même que décrit précédemment. Le vecteur en clair X_i est lu depuis la fin vers le début et le vecteur d'Huffman est lu du début vers la fin. Ceci construit la qualité de l'image d'une façon réglable.

Décryptage d'une région d'intérêt

Le décryptage d'une ROI peut être total ou partiel. Un des avantages de notre méthode est la possibilité de décrypter partiellement et de manière individuelle les blocs

8×8 pixels de l'image. Ceci est dû à deux faits : l'utilisation du mode par flot CFB ; la taille standard des blocs du JPEG 8×8. La région de l'image qui est à décrypter doit être définie dans des tailles de blocs unitaires de pixels 8×8.

5.5 Résultats

Pour toutes nos expériences, nous avons utilisé l'algorithme JPEG avec le système de codage en ligne séquentiel avec un facteur de qualité (FQ) de 100%. Nous avons appliqué sur toutes les images cinq valeurs pour la contrainte $C = (128, 64, 32, 16$ et $8)$. Pour le chiffrement, nous avons employé l'algorithme AES avec le mode de chiffrement par flot CFB avec une clef et un bloc de données de longueur 128 bits. Cependant, notre méthode peut être employée avec d'autres valeurs de longueur de clef acceptée par AES (à savoir 192 et 256 bits).

Nous avons appliqué notre méthode à trois domaines différents : à l'imagerie médicale pour le télédiagnostic, à une BDD de peintures numériques et à la vidéosurveillance.

5.5.1 Application à l'imagerie médicale

Les méthodes ont été appliquées sur plusieurs dizaines d'images médicales en niveau de gris [PRDM06, RPDM06]. Nous présentons les résultats tableaux 5.4 et 5.5 pour deux images médicales différentes, illustrées figures 5.6 et 5.7.

	Information cryptée				
C	Coeffi cients	Bits	% Bits	% pixels changés	PSNR (dB)
128	26289	81740	23,0	85,71	23,39
64	23987	71900	20,2	85,67	24,42
32	18035	52101	14,6	85,3	25,02
16	10966	31106	8,8	83,5	27,66
8	6111	16765	4,7	76,1	30,90

TAB. 5.4 – Résultats pour l'image rayons X d'un cancer du colon, figure 5.6.a.

L'image médicale originale, de taille 320×496 pixels, comprimée ainsi que toutes les images cryptées ont la même taille, soit $43,4$ Ko. Pour $C = 128$, maximum de 128 bits chiffrés par bloc, nous avons eu 26289 coefficients AC chiffrés et 81740 bits chiffrés, ce qui fait une moyenne de 33 bits chiffrés par bloc. Le pourcentage de bits chiffrés dans l'image entière est de $23,0\%$ et $85,7\%$ des pixels chiffrés. Ceci nous donne dans le domaine spatial 136023 pixels changés. Le pic du rapport signal à bruit (PSNR) est de $23,39\ dB$. Nous constatons donc que la visibilité de l'image cryptée est fortement réduite pour une faible pourcentage de bits cryptés.

Pour $C = 8$ la quantité de coefficients AC et de bits codés est respectivement de 6111 et de 16765. Le pourcentage de bits chiffrés par rapport à l'image entière est de $4,70\%$.

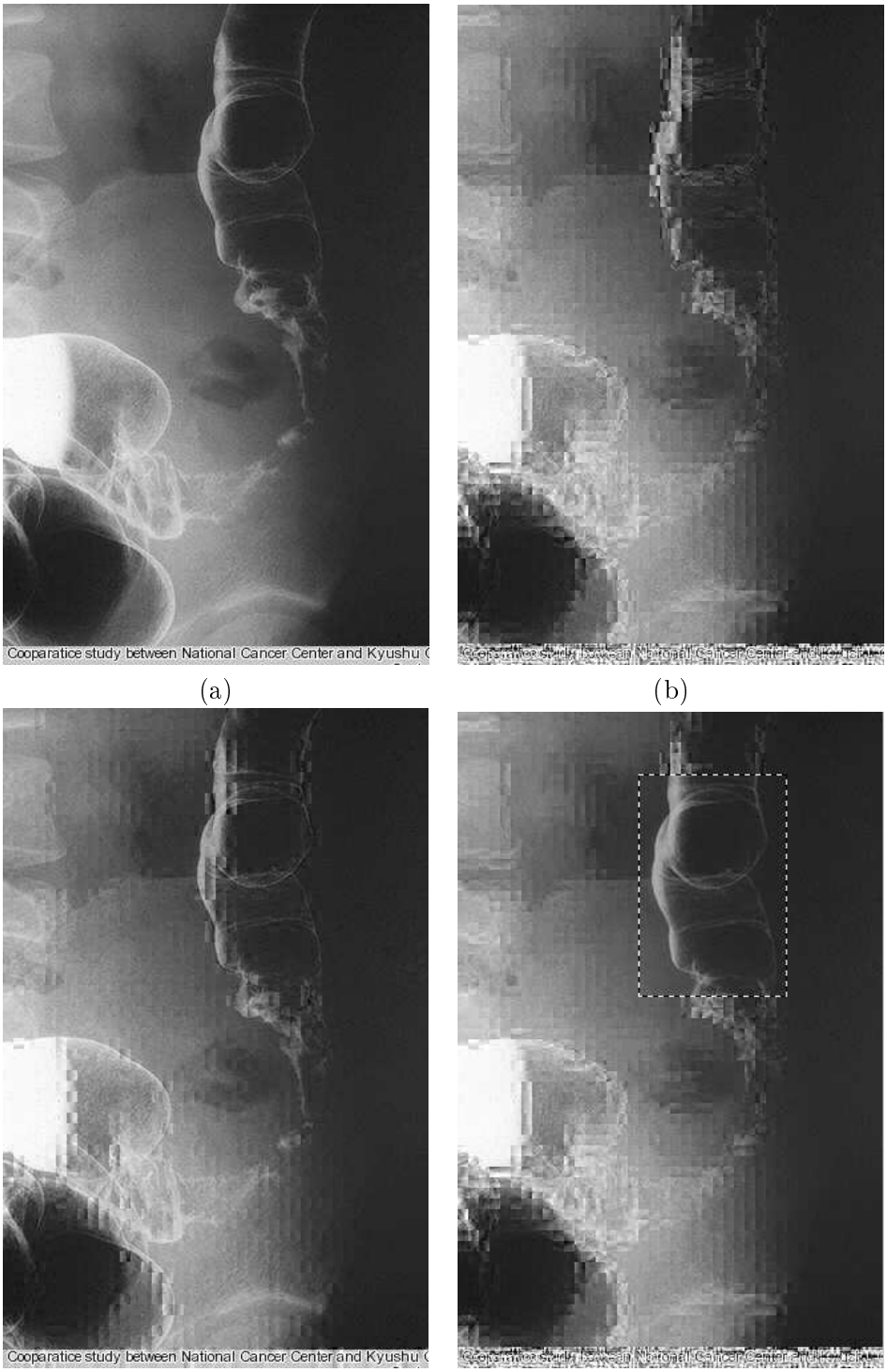
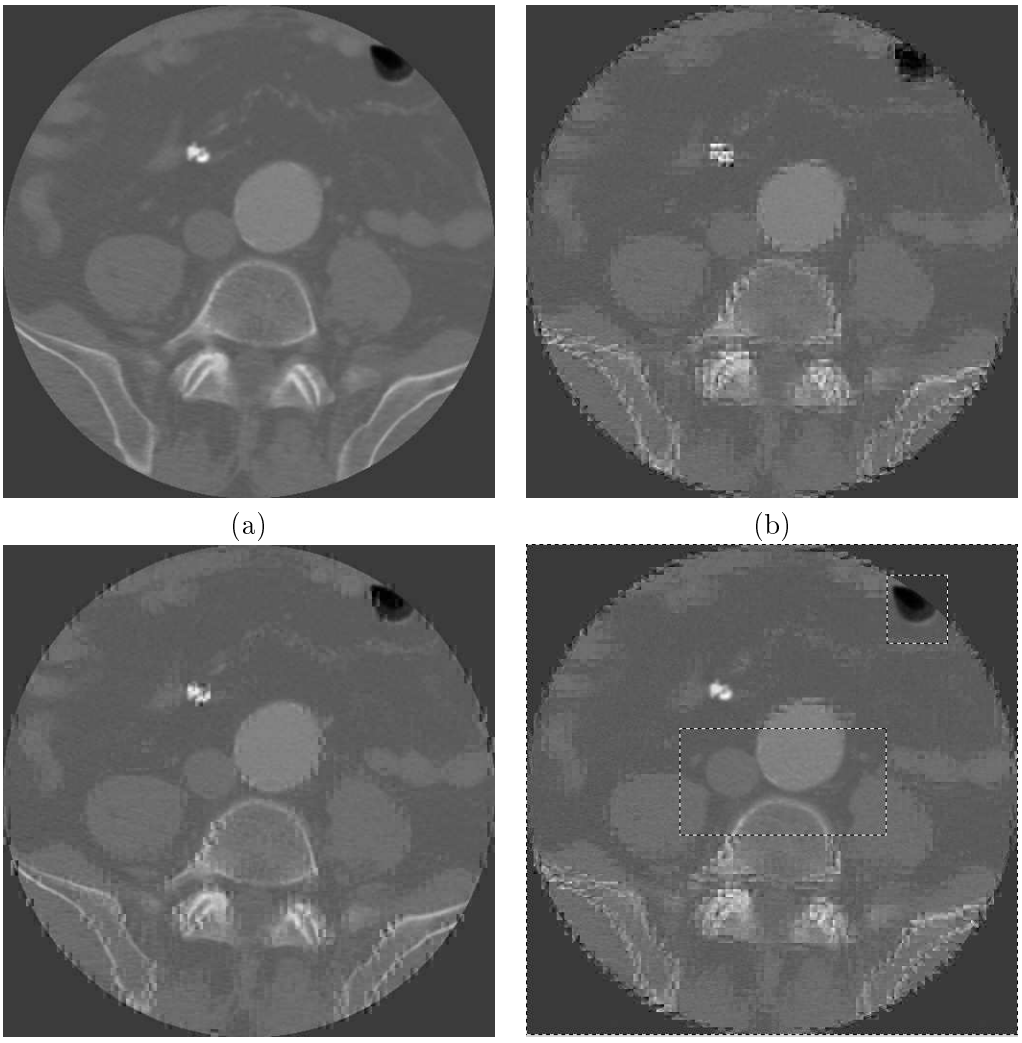


FIG. 5.6 – a) Image médicale originale d'un cancer du colon, b) Image cryptée pour $C = 128$, c) Image cryptée pour $C = 8$, d) Décryptage d'une région à 100%.

Cette contrainte nous donne $76,1\%$ de pixels modifiés qui correspond à un nombre de 120785 pixels dans toute l'image. Le PSNR est alors de $30,90\ dB$.

La figure 5.6.d montre le décryptage total sur une ROI de 13×9 blocs (soit 104×72 pixels).

Dans le tableau 5.5 nous montrons le résultat de notre méthode appliquée sur une image médicale d'un scanner CT de taille 512×512 pixels, figure 5.7. L'image originale après compression classique par JPEG et les images cryptées ont toutes la même taille, soit $59,9$ Ko.



(a) (b)

(c) (d)

FIG. 5.7 – a) Image médicale d'un scanner 512×512 pixels, b) Image cryptée pour $C = 128$, c) Image cryptée pour $C = 8$, d) Décryptage de deux régions à 50%.

Pour la contrainte $C = 128$, nous avons chiffré 51147 coefficients AC et 131127 bits,

ce qui correspond à 32 bits par bloc en moyenne. Le pourcentage de bits chiffrés dans l'image entière est de $26,7\%$, ce qui nous donne 230424 pixels changés, soit $87,9\%$ des pixels. Le PSNR est de $28,18\ dB$. Dans ce second exemple nous constatons également que la clarté de l'image est fortement réduite.

	Information cryptée				
C	Coeffi cients	Bits	% Bits	% pixels changés	PSNR (dB)
128	51147	131127	26,7	87,9	28,18
64	47656	119423	24,3	87,9	28,31
32	37995	95850	19,5	87,5	29,15
16	18957	53083	10,8	85,0	30,45
8	9633	26606	5,4	74,7	33,06

TAB. 5.5 – Résultats pour l'image médicale scanner CT, 512×512 pixels.

Pour $C = 8$, la quantité de coefficients et de bits chiffrés est respectivement de 9633 et de 26606. Seulement $5,4\%$ des bits de l'image sont chiffrés. Pourtant, avec $C = 8$, nous avons quand même 195769 des pixels de l'image qui sont modifiés, ce qui correspond à $74,7\%$ de tous les pixels. Le PSNR est alors de $33,06\ dB$.

La figure 5.7.d présente le décryptage partiel de deux ROIs. Dans cet exemple, les régions ont été décryptées à 50%. La première région est de taille 40×48 pixels pour celle située à gauche et de 64×64 pixels pour celle située à droite.

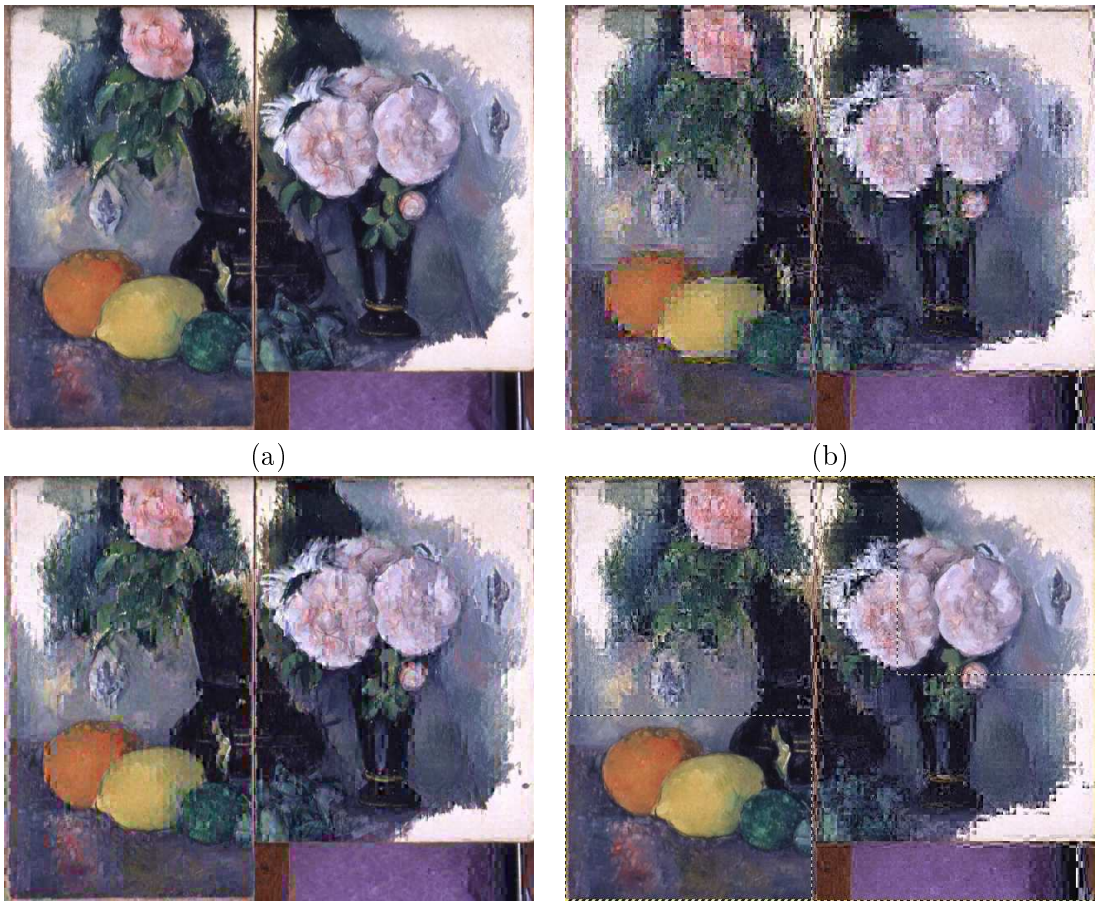
5.5.2 Application de notre méthode à une BDD de peintures numériques

La protection d'œuvres d'art numérisées pose encore des problèmes de sécurité quand elles sont mises en ligne. Donner des accès à différents niveaux de résolution d'une image en fonction des droits, masquage des données haute résolution est un intérêt du laboratoire C2RMF (Centre de Recherche et de Restauration des Musées de France, UMR CNRS 171) partenaire du projet TSAR (Transfert Sécurisé d'image numérique haute Résolution)[1]. L'objectif du projet TSAR est de transmettre de manière sécurisée des images haute résolution [RPB06a]. Une des stratégies de ce projet reposera sur l'utilisation de cryptage sélectif d'images et pour un transfert dans un temps raisonnable.

Nous avons donc appliqué notre méthode à la BDD de peintures numériques du C2RMF. Nous présentons les résultats tableaux 5.6 et 5.7 pour deux peintures, exposées figures 5.8 et 5.9.

Nous avons vu, chapitre 3 que pour les images couleur la norme JPEG réalise une transformation d'espace couleur RGB $\rightarrow$ YCbCr. Le Y est la luminance, Cb et Cr sont les chrominances. Notre méthode de cryptage sélectif a été mis en œuvre sur la composante Y seulement des peintures numériques. Les composantes de chrominances

[1] http ://www.lirmm.fr/tsar/



(a) (b)

(c) (d)

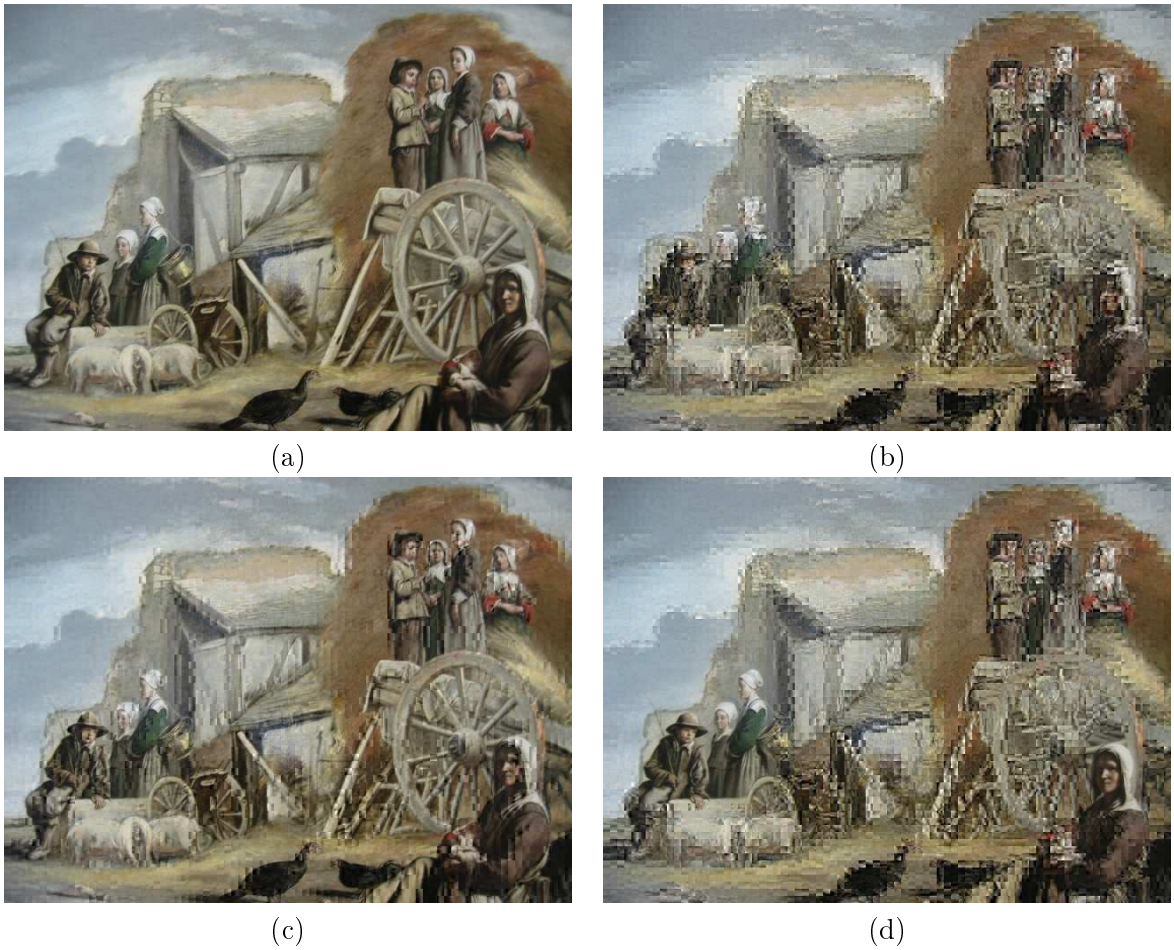
FIG. 5.8 – a) Peinture numérique originale 512×640 pixels, b) Image cryptée pour $C = 128$, c) Image cryptée pour $C = 8$, d) Décryptage de deux ROIs.

	Information cryptée				
C	Coeffi cients	Bits	% Bits	% pixels changés	PSNR (dB)
128	305033	733300	34,70	99,70	19,31
64	172902	449369	21,26	99,68	19,67
32	82399	238212	11,27	99,63	20,40
16	39735	121292	5,74	99,34	21,65
8	20805	60956	2,88	98,25	25,87

TAB. 5.6 – Résultats pour la peinture numérique figure 5.8.

ne sont pas utilisées parce qu'elles sont sous-échantillonnées dans la norme JPEG et elles ont beaucoup de coefficients DCT quantifiés à zéro.

Le tableau 5.6 expose le résultat de notre méthode appliquée à l'œuvre d'art figure 5.8. L'image originale après compression JPEG et les images cryptées ont toutes la même taille, soit 258 Ko. Pour la contrainte $C = 128$, nous avons chiffré 305033 coefficients AC et 733300 bits. Le pourcentage de bits chiffrés dans l'image entière est de $34,70\%$, ce qui nous donne 326686 pixels changés, soit $99,70\%$ des pixels. Le PSNR est de $19,31\ dB$.



(a) (b)

(c) (d)

FIG. 5.9 – a) Peinture numérique originale 640×480 pixels, b) Image cryptée pour $C = 128$, c) Image cryptée pour $C = 8$, d) Décryptage d'une région.

Sur le tableau 5.7, nous pouvons faire les mêmes observations que précédemment. Cependant, pour les œuvres d'art, il est important remarquer que presque 100% des pixels ont été modifiés. Ce phénomène est dû à la variation de couleurs entre les pixels. En effet dans les peintures numériques même si des régions semblent homogènes, les

	Information cryptée				
C	Coeffi cients	Bits	% Bits	% pixels changés	PSNR (dB)
128	168287	495458	20,2	95,5	20,27
64	99858	305994	12,5	95,3	20,39
32	46092	156473	6,4	94,9	21,03
16	21897	78119	3,2	93,4	22,38
8	11808	38864	1,6	89,7	26,42

TAB. 5.7 – Résultats pour la peinture numérique figure 5.9.

pixels voisins sont très souvent différents. Rappelons que dans le cas des images médicales les pixels des zones homogènes possèdent presque tout le temps la même valeur.

5.5.3 Application de notre méthode à la protection des visages dans des séquences d'images

Nous avons appliqué notre méthode à des séquences d'images (640 × 480 pixels) générées par une caméra fixe de surveillance [RPM+06]. L'objectif est de protéger les visages des personnes prises par la caméra fixe afin de résoudre le problème de protection de la vie privée et des libertés.

Les explications présentées sections 5.5.1 et 5.5.2 sont des cryptages de toute l'image et éventuellement un décryptage d'une ROI. Dans le cas de la protection des visages dans une séquence d'images, nous employons notre méthode pour crypter uniquement des ROIs dans l'image. Ces ROIs contiennent les visages des personnes passant devant la caméra de surveillance.

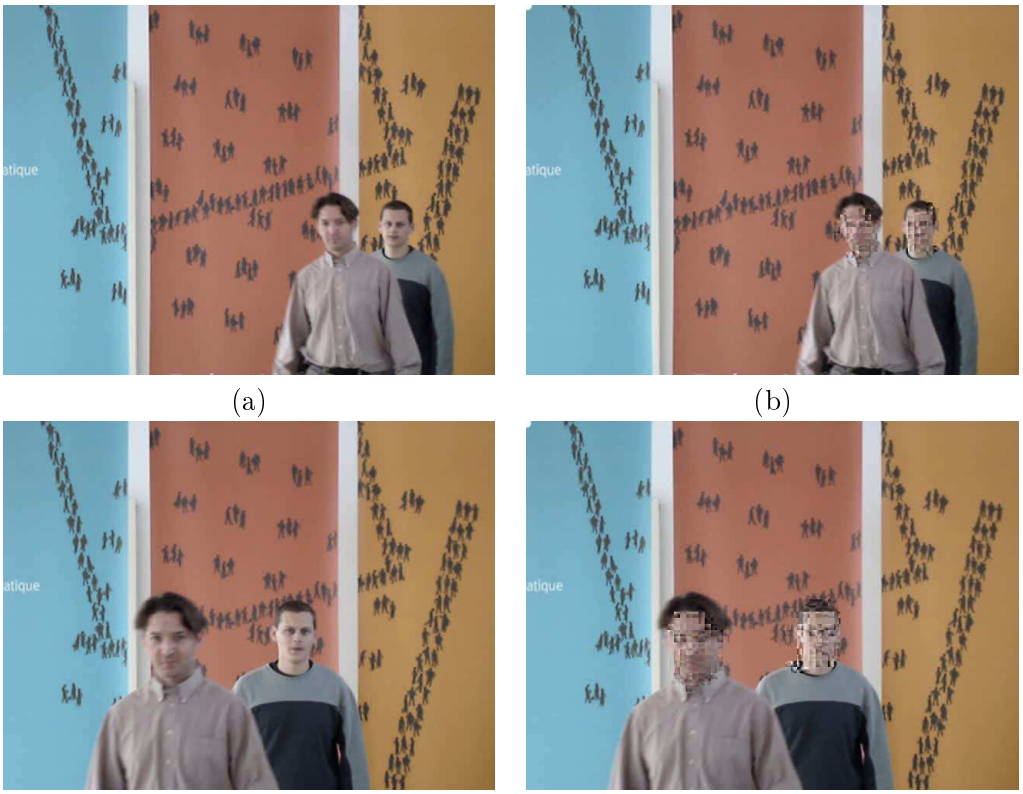
Les ROIs à crypter peuvent être choisies d'une façon manuelle par l'utilisateur, avant d'envoyer les images par le réseaux. Les régions à crypter peuvent être détectées également d'une façon automatique à l'aide des divers algorithmes de détection de visage existantes.

Rodrigues *et al.* [RPB06b] ont proposé une approche colorimétrique pour la détection de la peau/visage. Il s'agit de prendre les coefficients DC des composantes de crominances Cb et Cr pour produire deux petites images (8 fois plus petites) qui seront employées pour détecter la peau humaine. Sur ces petites images nous calculons la distance Euclidienne de chaque point par rapport à un seuil donné.

Comme le présente la figure 5.11, nous avons agrandi une région des visages afin de montrer clairement nos résultats. Les figures 5.11.a et b présentent des régions de 216 × 152 prises des figures 5.10.c et d respectivement.

5.6 Bilan

Comme nous pouvons voir sur les images résultats, le cryptage sélectif sur toute l'image JPEG produit des artefacts par bloc. Ces artefacts sont au niveau des frontières



(a) (b)

(c) (d)

FIG. 5.10 – a) Image originale #083 de la séquence, b) Image (a) cryptée pour $C = 128$, c) Image originale #135 de la séquence, d) Image (c) cryptée pour $C = 128$.

(a) (b)

FIG. 5.11 – a) Région de l'image figure 5.10.c, b) Région de l'image figure 5.10.d.

des blocs, qui importunent souvent le SVH. Puisque la transformation fréquentielle et la quantification des blocs de pixels sont traitées séparément, la continuité des valeurs des pixels de blocs voisins est cassée durant le codage.

Notre approche a incorporé des améliorations concernant l'approche de Droogenbroeck et Benedett [DB02]. Nous remarquons que :

1. Nous avons un gain de temps de calcul significatif par rapport leur approche. Dans notre démarche le cryptage sélectif est fait en même temps que la compression, tout en conservant le taux de compression et la compatibilité avec le format JPEG. De plus, nous nous servons de l'algorithme AES qui est beaucoup plus rapide que les algorithmes (DES, triple-DES et IDEA) [DH77] utilisés par Droogenbroeck et Benedett.
2. L'idée de chiffrer en partant des hautes fréquences (détails de l'image) et en terminant par les plus basses fréquences a apporté un réglage visuel autant pour le cryptage que pour le décryptage.
3. Le cryptage de blocs séparément à l'aide d'un chiffrement par flot a rendu possible le traitement des régions d'intérêt. Nous avons montré par exemple le cryptage d'une ROI dans une image claire, figure 5.10 et le décryptage des ROIs dans une image déjà cryptée sélectivement, figures 5.9.d.
4. L'utilisation conjoint de réglage et le traitement de région d'intérêt a rendu possible l'application de notre méthode dans plusieurs domaines.

Néanmoins, il convient de noter que la sécurité est liée à la capacité de deviner les valeurs des données chiffrées (cryptanalyse). Par exemple, d'un point de vue de la sécurité, il est préférable de chiffrer les bits qui semblent les plus aléatoires. En effet, en pratique, le remplacement des valeurs des coefficients AC non nuls est plus difficile que les valeurs des coefficients de DC d'une image JPEG qui sont fortement prévisibles.

Conclusion

Dans ce chapitre nous avons présenté une nouvelle approche de cryptage sélectif basée sur les travaux de Droogenbroeck et Benedett [DB02] pour les images comprimées au format JPEG. La combinaison du cryptage sélectif et de la compression permet de gagner du temps de calcul et de conserver le format JPEG. Nous avons appliqué notre approche dans plusieurs domaines : imagerie médicale, visualisation de peintures numériques et protection de visages dans des séquences d'images.

Conclusion Générale

Bilan du travail effectué

Au cours de cette thèse nous avons étudié deux problématiques liées à la protection des images. Le premier problème concerne le transfert sécurisé d'images dans des réseaux numériques en utilisant un codage hybride (cryptage, insertion de données cachées et compression). L'autre problème étudié considère la visualisation et la protection réglable (*scalable*) des images au format JPEG. Nous avons développé trois algorithmes pour faire face à ces problèmes. D'un point de vue expérimental, nous avons implémenté nos méthodes sur une plate-forme PC windows et linux. Le grand nombre d'expérimentations effectuées nous a permis de valider nos méthodes. Certains éléments de ces algorithmes sont opérationnels sur le site du projet ICAR[2] du LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier), ce qui permet à l'utilisateur de s'en servir en temps réel.

Après avoir étudié un panel assez diversifié des techniques de cryptage, d'IDC et de compression d'images, et avoir analysé les problèmes de sécurisation du transfert d'images, nous avons élaboré nos approches de codage hybride.

Notre **première contribution** a porté sur la création d'une nouvelle technique qui prend en compte : la compression sans perte par décomposition et le codage par plage ; l'insertion de données cachées additive et le cryptage par mélange (*scrambling*). L'utilisation de codage hybride avec trois axes de recherche pour la sécurisation du transfert d'images est une stratégie très innovante et originale. Cette stratégie est prometteuse en raison de la croissance exponentielle du trafic des images sur les réseaux numériques. La phase d'analyse des résultats a permis de montrer que les images soumises à cette méthode ont une entropie optimale, une capacité d'insertion pour l'IDC convenable et un cryptage sélectif satisfaisant dans un environnement totalement sans perte d'information. Ensuite, nos efforts se sont naturellement orientés vers notre **deuxième contribution**. L'algorithme de transfert autonome qui utilise aussi l'approche de codage hybride. Cette approche repose sur l'utilisation de plusieurs catégories de cryptage (asymétrique, symétrique et par flot asynchrone) et aussi l'usage de l'IDC par substitution dans le domaine spatial. Les résultats ont montré que malgré que nous ayons utilisé le marquage substitutif les pertes d'informations sont acceptables. Nous avons appliqué

[2] http ://www.lirmm.fr/icar/

notre méthode sur des images réelles et sur des images médicales.

Notre **troisième contribution** repose sur l'amélioration d'une méthode de cryptage sélectif. Afin d'obtenir un modèle pour un environnement vaste et standard nous nous sommes penchés sur le plus populaire des formats d'image, le JPEG. La décision de travailler dans les phases du JPEG, plus précisément dans le codage entropique rend notre approche efficace et applicable au dispositif de faible puissance. Nous avons montré des résultats pour l'imagerie médicale, les peintures numériques et les images de surveillance. Nous avons ensuite optimisé cette méthode pour le cryptage et décryptage de régions d'intérêt.

Enfin, nous avons illustré les apports de toutes les méthodes développées sur des cas réels qui évoque l'intérêt de leurs utilisations dans la vie quotidienne.

Bien que les approches proposées soient relativement performantes, elles ne sont pas suffisantes pour réaliser une protection complètement sûre, parce que l'objectif a été de montrer l'utilisation conjointe des trois domaines. En effet, l'utilisation d'opérations de mélange dans la première méthode, l'algorithme de cryptage par flot asynchrone proposé dans la deuxième méthode et le cryptage d'une partie des informations dans la troisième méthode (cryptage sélectif) ne rendent pas nos méthodes robustes aux attaques.

Perspectives

Par ailleurs, de nombreuses pistes sont possibles pour améliorer et développer des nouvelles solutions.

- Pour la première méthode proposée, nous pouvons envisager d'essayer plusieurs parcours. Par contre, l'augmentation du nombre de parcours diminue l'efficacité (en temps) de la méthode. Au lieu de décomposer l'image en deux plans de 4 bits chacun, nous pouvons aussi envisager une analyse statistique avant la décomposition pour définir le nombre de bits de chaque plan.

- Concernant la deuxième méthode proposée nous pouvons envisager de marquer l'image cryptée dans des régions spécifiques, ou des régions déterminées pour l'utilisateur. Nous pouvons également utiliser des méthodes de marquage (*watermarking*) sans perte pour insérer la clef cryptée dans l'image cryptée. Il serait souhaitable de développer des évaluations sur la robustesse concernant les attaques. Nous souhaitons faire des études concernant la cryptanalyse de notre méthode pour classifier son niveau de sécurité face aux attaquants. Le travail de cryptanalyse pourra être faite conjointement avec l'équipe ARITH du LIRMM spécialiste en cryptanalyses.

- Pour le cryptage sélectif des images au format JPEG, une tentative intéressante est le changement de la contrainte C par rapport l'homogénéité de l'image. Cette contrainte peut être variable selon l'image et atteindre la taille maximale de la clef (256 bits) pour l'algorithme AES. Une autre suggestion pour l'optimisation du résultat du cryptage sélectif est l'exploitation du voisinage (l'échange de blocs voisins de 8x8 pixels). Notre approche de CS est certainement extensible aux séquences vidéo utilisant un codage de Huffman.

- Pour l'analyse de la qualité visuelle des images, nous envisageons d'utiliser une métrique de mesures perceptuelles des distorsions. Une analyse détaillée sur les images décryptées et sur les images cryptées sélectivement. En particulier, nous distinguons les travaux menés dans le laboratoire IRCCyN (Institut de Recherche en Communications et en Cybernétique de Nantes).

- Enfin, nous envisageons d'optimiser les algorithmes développés pour les rendre

efficaces dans des environnements de faible puissance comme par exemple l'échographie à distance. L'un des premiers axes d'optimisation consisterait à améliorer la portabilité des algorithmes pour l'application réelles client/serveur dans le cadre du Projet TSAR (Transfert Sécurisé d'image d'art haute Résolution) et pour des vidéo médicales.

Annexe A

Transformée cosinus discrète

La Transformée en cosinus discrète ou DCT-(*Discrete Cosine Transform*) convertit des données spatiale en une répresentation féquentielle groupée par niveaux d'intensité. La DCT traduit l'information contenue dans le bloc de l'image en une somme de fontions de référence ponderée par des coefficients réels. A partir des intensités des pixels $p(i,j)$, nous obtenons les coefficients DCT associés $F(u,v)$, équation (A.1). Le premier de ces coefficients DC (la composante continue) répresente la quantité d'information la plus importante.

$$F(u,v) = \frac{2}{n}C(u)C(v)\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}p(i,j)cos\left(\frac{\pi(2i+1)u}{2n}\right)cos\left(\frac{\pi(2j+1)v}{2n}\right) \qquad \text{(A.1)}$$

avec :

$$\begin{cases} C(x) = \frac{1}{\sqrt{2}} \text{ si } x = 0 \\ \\ C(x) = 1 \text{ si } x \neq 0 \end{cases},$$

où les variables i et j répresent les indices de ligne et de colone dans la matrice des données. Les u et v répresent les indices de ligne et de colonnes des coeficients transformés. Les deux cosinus définissent la fréquence du signal respectivement dans chacune des deux directions du plan de l'image. Les coefficients DCT, $F(u,v)$, définissent l'amplitude des signaux. Pour limiter leur grandeur, les intensités des pixels définies entre 0 et 255, sont centrées autour de 0, entre -128 et 127.

Le coefficient DCT, $F(0,0)$ est appelé composante continue ou composante DC (*Direct Current*). Il possède la propriété d'être proportionnel à la moyenne des intensités des pixels sur le bloc considéré. Les autres coefficients sont appelés AC (*Alternating Current*). Sur les coefficients ACs, les énergies sont groupées en basse, moyenne et haute fréquences.

L'équation A.2 présente la Transformée en Cosinus Discrète Inverse qui permet de revenir au domaine spatial à partir des coefficients DCT.

$$p(i,j) = \frac{2}{n} \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} C(u)C(v)F(u,v)cos\left(\frac{\pi(2i+1)u}{2n}\right) cos\left(\frac{\pi(2j+1)v}{2N}\right), \qquad \text{(A.2)}$$

Annexe B

Entropie

L'entropie d'une source $\mathcal{X}$ est une mesure de la quantité d'information moyenne par symbole pour répresenter de façon univoque toute réalisation possible.

Une image représente une suite de $M \times N$ variables aléatoires appelés pixels. Soient X_i tous les niveaux de gris possibles d'un pixel particulier P, et $p(X_i)$ la probabilité du niveau de gris X_i. La quantité d'information de X_i, notée $I(X_i)$, est telle que :

$$I(X_i) = log_2\left(\frac{1}{p(X_i)}\right) \tag{B.1}$$

L'entropie de P, notée $H(P)$, est la quantité d'information moyenne $I(X_i)$, elle est exprimée en bits/symbole.

$$H(P) = \sum_{i=0}^{255} p(X_i).I(X_i) \tag{B.2}$$

L'entropie est une mesure toujours positive. Elle est maximale lorsque X suit une loi uniforme. Plus la probabilité est petite, plus la quantité d'information est grande. Elle est nulle lorsque X est connue de manière certaine, c'est-à-dire si $p(X_i) = 1$ alors $I(X_i) = 0$.

L'entropie a un rôle important en traitement d'images. L'image est un ensemble, *a priori*, désordonné d'informations et son entropie est généralement élevée. Dans l'étape de compression, les transformations appliquées sur les images (ondelettes et TCD par exemple) ont pour objectif de diminuer l'entropie du signal, pour rendre l'information plus dense. Ces transformations ont pour but traiter plus facilement l'information *a posteriori* dans le codage entropique qui va représenter les données sur un nombre réduit de niveaux discrets.

Concernant le cryptage nous considérons qu'un système cryptographique est parfait si le message crypté (Y_i) est une variable uniforme et indépendante du texte clair (Xi) et de la clef K. Notons que, ceci est la définition dans le cas d'une entropie maximale, et qu'en pratique le message et la clef ne sont jamais totalement aléatoires.

Bibliographie

[AARAS99] A. M. Alattar, G. I. Al-Regib, and S. A. Al-Semari. Improved Selective Encryption Techniques for Secure Transmission of MPEG Video Bit-Streams. In *ICIP 99, International Conference in Image Processing, IEEE*, volume 4, pages 256–260, 1999.

[AGB03] F. Autrusseau, J.P. Guédon, and Y. Bizais. Mojette cryptomarking scheme for medical images. In *SPIE Medical Imaging*, volume 5032, pages 958–965, May 2003.

[AI04] H. Alasady and M. Ibnkahla. A Simple Data Pre-Distortion Technique for Satellite Communications : Design and Implementation on Altera DSP Board. Technical report, GSPx'04, 2004.

[AP05] P. Amat and W. Puech. Transfert sécurisée d'une ROI sans perte par une méthode d'insertion de données cachées robuste à la compression JPEG. In *GRETSI*, Louvain-La-Neuve, Belgique, Septembre 2005.

[APFM01] Parisis A., Carré P., and Fernandez-Maloigne. Watermarking et couleur : étude de différents espaces de représentation couleur. In *Coresa01*, Dijon, France, 2001.

[Aut02] F. Autrusseau. *Tatouage d'images basé sur la modélisation du Système Visuel Humain.* PhD thesis, Université de Nantes et Ecole Centrale de Nantes, 2002.

[Bas00] P. Bas. *Méthodes de tatouage d'images fondées sur le contenu.* PhD thesis, Institut National Polytechnique de Grenoble, 2000.

[BB04] M. Barni and F. Bartolini. *Watermarking Systems Engineering - Signal Processing and Communication Series.* Marcel Dekker, Inc., New York, USA, 2004.

[BC00] P. Bas and B. Chassery, J. M.and Macq. Robust Watermarking Based on the Warping of Pre-Defined Triangular Patterns. *Proc.SPIE Electronic Imaging, Security and Watermarking of Multimedia Contents II*, 3971 :99–110, 2000.

[BCM02] P. Bas, J. Chassery, and B. Macq. Geometrically invariant watermarking using feature points. *IEEE Trans. Image Processing*, 11 :1014–1028, 2002.

[BDR03] M. Babel, O. Déforges, and J. Ronsin. Décomposition pyramidale à redondance minimale pour compression d'images sans perte. In *GRETSI*, volume 1, page CD–ROM, 2003.

[BGM96] W. Bender, D. Gruhl, and N. Morimoto. Techniques for data hiding. *IBM Systems Journal*, 35 :131–336, 1996.

[BI04] J. C. Bajard and L. Imbert. A Full RNS Implementation of RSA. *IEEE Transactions on Computers*, 53(6) :769–774, 2004.

[BINP03] J.-C. Bajard, L. Imbert, C. Negre, and T. Plantard. Efficient multiplication in GF(pk) for Elliptic Curve Cryptography. In *ARITH'03 Proceedings of the 16th IEEE Symposium on Computer Arithmetic*, pages 181–187, 2003.

[BL01] T. Berger and P. Loidreau. Weak keys in McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3) :1207–1212, March 2001.

[Blo85] R. Blom. An optimal class of symmetric key generation systems. In *EUROCRYPT 84*, volume 209, page 335–338, Springer-Verlag, 1985. Advances in Cryptology.

[BM84] M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudorandom Bits. *SIAM Journal on Computing*, 13(4) :850–864, 1984.

[Bor04] J. C. Borie. *Sécurisation d'images par cryptage : applications aux images médicales*. PhD thesis, Université de Nîmes, 2004.

[BPD+05] M. Babel, B. Parrein, O. Déforges, N. Normand, J.-P. Guédon, and J. Ronsin. Secured and progressive transmission of compressed images on the Internet : application to telemedicine. In *SPIE*, volume 5670, pages 126–136, December 2005.

[BRM03] P. Bas, B. Roue, and Chassery J. M. Tatouage d'images couleur additif : vers la sélection d'un espace d'insertion optimal. In *Coresa03*, volume 1, Lyon, France, 2003.

[BRW04] M. Bellare, P. Rogaway, and D. Wagner. The EAX Mode of Operation. In *FSE*, volume 3017 of *Lecture Notes in Computer Science*. Springer, 2004.

[BS04] Z. Brahimi and K.A. Saadi. Color image coding based on embedded wavelet zerotree and scalar quantization. In *International Conference on Pattern Recognition*, volume 1, pages 504–507, 2004.

[CCC02] Chin-Chen Chang, Tung-Shou Chen, and Lou-Zo Chung. A steganographic method based upon JPEG and quantization table modification. *Inf. Sci. Inf. Comput. Sci.*, 141(1-2) :123–138, 2002.

[CCGN03] P. Campisi, M. Carli, G. Giunta, and A. Neri. Blind quality assessment system for multimedia communications using tracing watermarking. *Signal Processing, IEEE Transactions*, 51(4) :996–1002, 2003.

[CCH05] C.-C. Chang, C.-C. Chiang, and J.-Y. Hsiao. A DCT-Domain System for Hiding Fractal Compressed Images. In *AINA05*, volume 2, pages 83–86. IEEE Inter.Conf.on Advanced Information. Networking and Applications, 2005.

[CdV01] A. Canteaut and F. Lévy dit Véhel. La cryptologie moderne. *L'Armement*, 73 :76–83, 2001.

[CHC01] C. C. Chang, M.S. Hwang, and T-S Chen. A new encryption algorithm for image cryptosystems. *The Journal of Systems and Software*, 58 :83–91, 2001.

[CJM96] D. Coppersmith, D. Johnson, and S. Matyas. A proposed mode for triple-DES encryption. *IBM Journal of Research and Development*, 40(2) :253–262, 1996.

[CKM94] D. Coppersmith, H. Krawczyk, and Y. Mansour. The shrinking generator. In *CRYPTO '93 : Proceedings of the 13th annual international cryptology conference on Advances in cryptology*, pages 22–39, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[CL00] H. Cheng and X. Li. Partial Encryption of Compressed Images and Videos. *IEEE Transactions on Signal Processing*, 48(8) :2439–2451, 2000.

[CLC+01] D.R. Clewer, L.J. Luo, C.N. Canagarajah, D.R. Bull, and M.H. Barton. Efficient multiview image compression using quadtree disparity estimation. In *ISCAS - IEEE Int. Symp. on Circuits and Systems*, volume 5, pages 295–298, 2001.

[CLD+05] G. Coatrieux, M. Lamard, W. Daccache, J. Puentes, and C. Roux. A low distortion and reversible watermark : Application to angiographic images of the retina. In EMBC'05, editor, *Proceedings of Int. Conf. of the IEEE-EMBS*, pages 2224–2227, Shangaï, China, November 2005.

[CMB02] I. Cox, M. Miller, and J. Bloom. *Digital watermarking*. Morgan Kaufmann Publishers Inc., San Fransisco, USA, 2002.

[CPS03] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, page 197–213, 2003.

[CW01] B. Cheng and G. W. Wornell. Quantization Index Modulation : A Class of Provably Good Methods for Digital Watermarking and Information Embedding. *IEEE transaction on information theory*, 47(4) :1423–1443, 2001.

[DB02] M. Van Droogenbroeck and R. Benedett. Techniques for a Selective Encryption of Uncompressed and Compressed Images. In *ACIVS - Advanced Concepts for Intelligent Vision Systems*, 2002.

[DD97] Kundur D. and Hatzinakos D. A Robust Digital Image Watermarking Scheme Using the Wavelet Based Fusion. In *IEEE-ICIP'97*, volume 1, pages 544–547, 1997.

[DFHS03] J. Delhumeau, T. Furon, N. Hurley, and G. Silvestre. Improved Polynomial Detectors for Side-Informed Watermarking. In *Proc. of Security and Watermarking of Multimedia Contents V, SPIE Electronic Imaging*, Santa Clara, CA, USA, January 2003.

[DH76] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6) :644–654, 1976.

[DH77] W. Diffie and M. Hellman. Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *IEEE Computer*, 10(6) :74–84, 1977.

[Dje06] Habib Djebali. Analyse et Cryptanalyse d'un système de chiffrement d'images. Master's thesis, Université Montpellier II - LIRMM, 2006.

[DM00] D. Delanay and B. Macq. Generalized 2-D cyclic patterns for secret watermark generation. In *IEEE ICIP*, volume 2, page 77–80, 2000.

[DP03] F. Davoine and S. Pateux. *Tatouage de documents audiovisuels numériques*. Hermès Science Publications, Lavoisier, France, 2003.

[DR02a] J. Daemen and V. Rijmen. AES Proposal : The Rijndael Block Cipher. Technical report, Proton World Int.l, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, 2002.

[DR02b] J. Daemen and V. Rijmen. *The Design of Rijndael*. SpringerVerlag New York, Inc. Secaucus, NJ, USA, 2002.

[DW05] S. Dumitrescu and X. Wu. LSB steganalysis based on high-order statistics. In *MM&Sec '05 : Proceedings of the 7th workshop on Multimedia and security*, pages 25–32, New York, NY, USA, 2005. ACM Press.

[EBTG03] J. Eggers, R. Buml, R. Tzschoppe, and B. Girod. Scalar Costa Scheme for Information Embedding. *IEEE Transactions on Signal Processing, Special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery*, 2003.

[EE06] E. Elbasi and A. M. Eskicioglu. A Semi-Blind Watermarking Scheme for Images Using a Tree Structure. In *IEEE Sarnoff Symposium*, March 2006.

[ES04] Ö. Ekici and B. Sankur. Comparative Evaluation of Semifragile Watermarking Algorithms. *Journal of Electronic Imaging*, 13(1) :209–216, January 2004.

[FB98] Hartung F. and Girod B. Watermarking of Uncompressed and Compressed Video. *Signal Processing*, 66(3) :283–333, 1998.

[FD01] J. Fridrich and G. Dur. Invertible authentication. In *SPIE*, San Jose, California, 2001. Security and watermarking of multimedia contents.

[FGCP04] J. Fridrich, M. Goljan, Q. Chen, and V. Pathak. Lossless data embedding with file size preservation. In *Security, Steganography, and Watermarking of Multimedia Contents*, pages 354–365, 2004.

[FGS05] J. Fridrich, M. Goljan, and D. Soukal. A New Steganographic Method for Palette-Based Images. In *CCCC*. 43rd Conference on Coding, Communication, and Control, September 2005.

[FHF02] B. Fong, G.Y. Hong, and A.C.M Fong. Constrained error propagation for efficient image transmission over noisy channels. *IEEE Transactions on Consumer Electronics*, 48(1) :49–55, 2002.

[FPPJM99] Davoine F., Bas P., Hebert P., and Chassery J.-M. Watermarking et Résistance aux déformations géométriques. In *Coresa*, juin 1999.

[Fri99] J. Fridrich. A New Steganographic Method for Palette-Based Images. In *PICS*, Georgia, USA, April 1999. IS&T - The Society for Imaging Science and Technology.

[FSU04] M. M. Fisch, H. Stgner, and A. Uhl. Layered Encryption Techniques for DCT-Coded Visual Data. In *(EUSIPCO) European Signal Processing Conference*, Vienna, Austria, Sep., 2004.

[GFG01] A. Guyader, E. Fabre, and C. Guillemot. Joint source-channel turbo decoding of VLC encoded Markov sources. In *GRETSI*, pages CD–ROM, septembre 2001.

[GG92] A. Gersho and R. M. Gray. *Vector quantization and signal compression.* Kluwer, Boston, 1992.

[GM84] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28 :270–299, 1984.

[GM03] L. Guillemot and J-M. Moureaux. Tatouage d'images : une nouvelle approche basee sur une methode de compression. In *CORESA*, pages 253–256, Lyon, France, 2003.

[GP03] Le G. Gaëtan and S. Pateux. Wide Spread Spectrum Watermarking with Side Information and Interference Cancellation. In *Proceedings of SPIE*, Santa Clara, CA, U.S.A, January 2003.

[GTOMD05] G.S.El-Taweel, H.M. Onsi, M.Samy, and M.G. Darwish. Secure and Non-Blind Watermarking Scheme for Color Images. *ICGST International Journal on Graphics, Vision and Image Processing*, SI1, 2005.

[HHKC03] Yongjian Hu, Jiwu Huang, Sam Kwong, and Yiu-Keung Chan. Image Fusion Based Visible Watermarking Using Dual-Tree Complex Wavelet Transform. In *IWDW*, pages 86–100, 2003.

[HJRS01] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel. Lossless Recovery of an Original Image Containing Embedded Data. In *US Pat. 6,278,791*, 2001.

[Huf52] D. A. Huffman. A Method of the Construction of Minimum Redundancy Codes. In *IRE*, volume 40, pages 1098–1101, 1952.

[HV94] Paul G. Howard and Jeffery Scott Vitter. Arithmetic Coding for Data Compression. Technical Report Technical report, DUKE–TR–1994–09, 1994.

[HW99] C. T. Hsu and J. L. Wu. Hidden Digital Watermarks in Images. *IEEE Transactions on Image Processing*, 8(1) :58–68, 1999.

[IdSC06] Y. Iano, F.S. da Silva, and A.L.M. Cruz. A fast and efficient hybrid fractal-wavelet image coder. *IEEE Transactions on Image Processing*, 15(1) :98–105, 2006.

[JA03] P. Justin and R. Arnaud. Optimizing watermark robustness with respect to a perceptual distortion constraint. *Proceedings of SPIE - Security and Watermarking of Multimedia Contents V*, 5020 :115–122, 2003.

[JD03] H. Joumaa and F. Davoine. Tatouage substitutif d'images intégrant un masque de pondération visuelle. In *CORESA*, Lyon, France, Jan 2003.

[JGB96] Kyung Sub Joo, D. R. Gschwind, and T. Bose. ADPCM encoding of images using a conjugate gradient based adaptive algorithm. In *ICASSP - IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 4, pages 1942–1945, 1996.

[JJ98] N. F. Johnson and S. Jajodia. Exploring Steganography : Seeing the Unseen. *IEEE Computer*, 31(2) :26–34, 1998.

[JM99] D. Johnson and A. Menezes. The Elliptic Curve Digital Signature Algorithm ECDSA, 1999.

[Kah92] D. Kahn. *The Histories - Terpsichore - Polymnia.* J.M. Dent & Sons Ltd, London England, 1992.

[Kah96] D. Kahn. *The Codebreakers - The Story of Secret Writing.* Scribner, New York, 1996.

[Ker83] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX :161–191, February 1883.

[KH05] L. Kamstra and H.J.A.M. Heijmans. Reversible Data Embedding Into Images Using Wavelet Techniques and Sorting. *IP*, 14(12) :2082–2090, December 2005.

[KKC03] Tae Young Kim, Taejeong Kim, and Hyuk Choi. Correlation-based asymmetric watermark detector. In *ITCC 2003*, pages 564 – 568, Las Vegas, NV, USA, 2003. IEEE - Inter. Conf.on Information Technology Coding and Computing.

[KMK03] M. Kurosaki, K. Munadi, and H. Kiya. Error correction using data hiding technique for JPEG2000 images. In *ICIP03*, pages 473–476, 2003.

[Kob90] M. Kobayashi. Digital Watermarking : Historical Roots. Technical report, IBM Research, Tokyo Research Laboratory, Japan, April 1990.

[KP00] S. Katzenbeisser and F. A. P. Petitcolas. *Information Hiding Techniques for Steganography and Digital Watermarking.* Artech House, London, 2000.

[Kut99] M. Kutter. Watermarking resisting to translation, rotation and scaling. In *SPIE, Multimedia systems and applications*, volume 3528, pages 423–431, 1999.

[LC98] C.-Y. Lin and S.-F. Chang. Robust Image Authentication Method Surviving JPEG Lossy Compression. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 296–307, 1998.

[LG05] G. Le Guelvouit. Trellis-coded quantization for public-key steganography. In *ICASP05*, Philadelphia, USA, Mars 2005. IEEE Conf.on Acoustics, Speech and Signal Processing.

[Li04] C.-T Li. Digital fragile watermarking scheme for authentication of JPEG images. *IEE Proceedings Vision, Image and Signal Processing*, 151(6) :460– 466, December 2004.

[Li05] C.-T Li. Reversible watermarking scheme with image-independent embedding capacity. *IEE Proceedings Vision, Image and Signal Processing*, 152(6) :779–786, December 2005.

[LJ87] H.W. Lenstra Jr. Factoring Integers with Elliptic Curves. *Annals of Mathematics*, 126 :649–673, 1987.

[Lou01] S. Loureiro. *Mobile Code Protection.* PhD thesis, ENST, EURECOM, January 2001.

[LR88] M. Luby and Ch. Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM Journal of Computing*, 17(2) :373–386, 1988.

[LTS+03] A. K. Lenstra, E. Tromer, A. Shamir, W. Kortsmit, B. Dodson, J. Hughes, and P. C. Leyland. Factoring Estimates for a 1024-Bit RSA Modulus. *Lecture Notes in Computer Science*, 2894 :55–74, January 2003.

[LV01] A. Lenstra and E. Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology*, 14 :255–293, 2001.

[LVM04] Tie Liu, R. Venkatesan, and M. K. Mihak. Scale-invariant image watermarking via optimization algorithms for quantizing randomized statistics. In *MM&Sec '04 : Proceedings of the 2004 workshop on Multimedia and security*, pages 124–132, New York, NY, USA, 2004. ACM Press.

[LVPD05] G. Lo-Varco, W. Puech, and M. Dumas. Content Based Watermarking for Securing Color Images. *Journal of Imaging Science and Technology*, 49(4) :450–458, 2005.

[LXF01] Y. Lim, C. Xu, and D. D. Feng. Web based image authentication using invisible Fragile watermark. In *CRPITS'11 : Proceedings of the Pan-Sydney area workshop on visual informtaion processing conference on Visual information processing*, pages 31–34, Darlinghurst, Australia, 2001. Australian Computer Society, Inc.

[MB01] S. S. Maniccam and N. G. Bourbakis. Lossless image compression and encryption using SCAN. *Pattern Recognition*, 34(6) :1229–1245, 2001.

[MC05] Yun He Minghua Chen. A fragile watermark error detection scheme for wireless video communications. *IEEE Transactions on Multimedia*, 7(2) :201–211, April 2005.

[McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *JPL DSN Progress Report*, 4244 :114–116, 1978.

[MN98] Makoto Matsumoto and Takuji Nishimura. Mersenne Twister : A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Trans. Model. Comput. Simul.*, 8(1) :3–30, 1998.

[Mon96] B.G. Monro, D.M. ; Sherlock. Optimal quantisation strategy for DCT image compression. In *IEE Proceedings Vision, Image and Signal Processing*, volume 143, pages 10–14, 1996.

[Mor95] N. Moreau. *Techniques de compression des signaux*. Masson, Paris, 1995.

[MRK00] Saraju P. Mohanty, K. R. Ramakrishnan, and Mohan S. Kankanhalli. A DCT Domain Visible Watermarking Technique for Images. In *IEEE International Conference on Multimedia and Expo (II)*, pages 1029–1032, 2000.

[MRRN04] Saraju P. Mohanty, N. Ranganathan, and K. Ravi Namballa. VLSI Implementation of Visible Watermarking for a Secure Digital Still Camera Design. In *Proceedings. 17th International Conference on VLSI Design*, pages 1063–1068, 2004.

[MSCS06] K. Maeno, Q. Sun, S.-F. Chang, and M. Suto. New Semi-Fragile Image Authentication Watermarking Techniques Using Random Bias and Nonuniform Quantization. *IEEE Transactions on Multimedia*, 8(1) :32–45, 2006.

[MTI86] T. Matsumoto, Y. Takashima, and H. Imai. On Seeking Smart Public-key Distribution Systems. *In Transactions of the IECE of Japan*, 69 :99–106, 1986.

[MvOV01] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography (5th edition)*. CRC Press LLC, Florida, EUA, August 2001.

[New40] B. Newman. *Secrets of German Espionage*. Robert Hale Ltd, London, 1940.

[NKE02] H. Niimi, M.and Noda, E. Kawaguchi, and R. O. Eason. Luminance Quasi-Preserving Color Quantization for Digital Steganography to Palette-Based Images. In *ICPR02*, volume 1, pages I :251–254, August 2002.

[NMC+06] A. Ninassi, O. Le Meur, P. Le Callet, Dominique Barba, and A. Tirel. Task impact on the visual attention in subjective image quality assessment. In *EUSIPCO-06*, Florence, Italy, 2006.

[NP99] N. Nikolaidis and I. Pitas. Digital Image Watermarking : An Overview. In *ICMCS*, volume 1, pages 1–6, 1999.

[NPP+03] R. Norcen, M. Podesser, A. Pommer, H.P. Schmidt, and A. Uhl. Confidential Storage and Transmission of Medical Image Data. *Computers in Biology and Medicine*, 33 :277–292, 2003.

[NSSK02] H. Noda, J. Spaulding, M.N. Shirazi, and E. Kawaguchi. Application of bit-plane decomposition steganography to JPEG2000 encoded images. *SPLetters*, 9(12) :410–413, December 2002.

[Pat98] S. Pateux. *Segmentation spatio-temporelle et codage orienté-régions de séquences vidéo basés sur le formalisme MDL.* PhD thesis, Université de Rennes 1, 1998.

[PBJM03] Bas P., Roue B., and Chassery J.-M. Tatouage d'images couleur additif : vers la sélection d'un espace d'insertion optimal. In *Coresa03*, Lyon, France, 2003.

[Per05] L. Perret. *Etude d'outils algébriques et combinatoires pour la cryptographie à clef publique.* PhD thesis, Université de Marne-la-Vallée, 2005.

[PHA05] D. H. PHAN. *Sécurité et efficacité des schémas cryptographiques.* PhD thesis, École polytechnique, 2005.

[PM92] W. B. Pennebaker and J. L. Mitchell. *JPEG : Still Image Data Compression Standard.* Kluwer Academic Publishers, USA, December 1992.

[PR04a] W. Puech and J. M. Rodrigues. A New Crypto-Watermarking Method for Medical Images Safe Transfer. In *EUSIPCO'04 - European Signal Processing Conference*, volume 1, pages 1481–1484, 2004.

[PR04b] W. Puech and J. M. Rodrigues. Sécurisation d'image par crypto-tatouage. In *CORESA'04 - 9th Colloque Compression et Représentation des Signaux Audiovisuels*, page 215–218, 2004.

[PR05a] W. Puech and J. M. Rodrigues. Crypto-compression d'images médicales par cryptage partiel des coeffcients dct. In *JSTIM - Journées Sciences, Technologies et Imagerie pour la Médecine*, pages 149–150, Nancy, France, 2005.

[PR05b] W. Puech and J. M. Rodrigues. Crypto-Compression of Medical Images by Selective Encryption of DCT. In *EUSIPCO'05 - European Signal Processing Conference*, Antalya, Turkey, 2005.

[PR05c] W. Puech and J. M. Rodrigues. *Transfert sécurisé et autonome d'images.* Brevet 123-01 déposé par le CNRS en collaboration avec la société SIGILLUM Technologies, Septembre 2005.

[PR06] W. Puech and J. M. Rodrigues. An Autonomous Encrypto Data Hiding Method for Image Safe Transfer. *Journal Signal Processing : Image Communication, Elsevier*, In revision, 2006.

[PRDM06] W. Puech, J. M. Rodrigues, and J. E. Develay-Morice. Transfert sécurisé d'images médicales par codage conjoint : cryptage sélectif par aes en mode par flot et compression jpeg. *Revue Traitement du signal (TS) Traitement du signal appliqué à la cancérologie, numéro spécial*, 23(5), september 2006.

[Pro01] N. Provos. Defending Against Statistical Steganalysis. In *USENIX Security Symposium*, pages 323–335, August 2001.

[RCG02] Richard E. Woods Rafael C. Gonzalez. *Digital Image Processing.* Addison-Wesley Pub Co, ISBN : 0201180758, Paris, January 2002.

[RK05] A. Ramalingam and S. Krishnan. Robust image watermarking using a chirp detection-based technique. *IEE Proceedings Vision, Image and Signal Processing*, 152(6) :771–778, December 2005.

[RM02] R. Radhakrishnan and N. D. Memon. On the security of the digest function in the SARI image authentication system. *IEEE Trans. Circuits Syst. Video Techn.*, 12(11) :1030–1033, 2002.

[Rog02] P. Rogaway. Authenticated-encryption with associated-data. In *CCS-9*, Proceedings of the 9th Annual Conference on Computer and Communications Security, page 98–107. ACM, 2002.

[RP06] J. M. Rodrigues and W. Puech. An Adaptable Invertible Encrypto-Data Hiding Method for Still Heterogeneous Images. *Journal of Real-Time Image Processing, Springer*, Submitted, 2006.

[RPB06a] J. M. Rodrigues, W. Puech, and A. G. Bors. A Selective Encryption for Heterogenous Color JPEG Images Based on VLC and AES Stream Cipher. In *CGIV - Third European Conference on Color in Graphics, Imaging and Vision*, Leeds, UK, 2006.

[RPB06b] J. M. Rodrigues, W. Puech, and A. G. Bors. Selective Encryption of Human Skin in JPEG Images. In *ICIP'06, IEEE International Conference on Image Processing*, 2006.

[RPDM06] J. M. Rodrigues, W. Puech, and J. E. Develay-Morice. A Scalable Selective Encryption for JPEG Medical Image Based on VLC and AES Stream Cipher. *IEEE Medical Imaging*, Submitted, 2006.

[RPF04] J. M. Rodrigues, W. Puech, and C. Fiorio. Lossless Crypto-Data Hiding in Medical Images Without Increasing the Original Size. In *MEDSIP'04 - 2nd International Conference on Advances in Medical Signal and Information Processing*, page 358–365, 2004.

[RPM+06] J. M. Rodrigues, W. Puech, P. Meuel, J.C. Bajard, and M. Chaumont. Human Face Protection with Fast Selective Encryption in a Video Sequence. In *IET - Crime and Security*, volume 1, pages 420–425, 2006.

[RRP04] J. M. Rodrigues, J. R. Rios, and W. Puech. SSB-4 System of Steganography using bit 4. In *WIAMIS04 - International Workshop on Image Analysis for Multimedia Interactive Services*, Lisbon, Portugal, 2004.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21 :120–126, 1978.

[Sai05] A. Said. Measuring the Strength of Partial Encryption Scheme. In *ICIP - IEEE International Conference in Image Processing*, volume 2, pages 1126–1129, 2005.

[SGS03] Gene Spafford, Simson Garfinkel, and Alan Schwartz. *Practical UNIX & Internet Security*. O'Reilly & Associates, inc, USA, 2003.

[Sha49] Claude E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4) :656–715, 1949.

[Sho97] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal of Computing*, 26(5) :1484–1509, 1997.

[Sie01] D. Sieberg. Bin Laden exploits technology to suit his needs. Technical report, CNN, New York, USA, September 2001.

[Sim83] G. J. Simmons. The Prisioners' Problem and the Subliminal Channel. *Advances in Cryptology, Proceeding of CRYPTO 83*, pages 51–67, 1983.

[Sin04] A. Singh. Eraser : An Exploit-Specifc Monitor to Prevent Malicious Communication Channels. In *SIGCOMM04*, Oregon, USA, September 2004. ACM SIGCOMM' 04.

[SJC+02] K. Solanki, N. Jacobsen, S. Chandrasekaran, U. Madhow, and B. S. Manjunath. High Volume Data Hiding in Images : Introducing Perceptual Criteria Into Quantization. In *ICASSP02*, May 2002.

[SkCjJyEy03] J. Sung-kwan, S. Chang-jin, L. Jin-young, and C. Eui-young. Self-organizing Coefficient for Semi-blind Watermarking. *Lecture Notes in Computer Science*, 2642 :275–286, January 2003.

[SLL01] B.-J. Shieh, Y.-S. Lee, and C.-Y. Lee. A new approach of group-based VLC codec system with full table programmability. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(2) :210–221, 2001.

[SMCM05] K. Sullivan, U. Madhow, S. Chandrasekaran, and B. S. Manjunath. Steganalysis of spread spectrum data hiding exploiting cover memory. In *IS&T/SPIE's 17th Annual Symposium on Electronic Imaging Science and Technology*, January 2005.

[SP05] Stefaan Seys and Bart Preneel. The Wandering Nodes : Key Management for Low-Power Mobile Ad Hoc Networks. In *ICDCSW'05*, pages 916–922, 2005.

[SS03] A. Sinha and K. Singh. A technique for image encryption using digital signature. *Optics Communications*, 218 :229–234, 2003.

[SSM03] T. Sohn, J. Seo, and J. Moon. A study on the covert channel detection of TCP/IP header using support vector machine. *Information and Communications Security*, 2836 :313–324, 2003.

[SSM+05] K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Statistical Restoration for Robust and Secure Steganography. In *IEEE International Conference on Image Processing*, Septembre 2005.

[Sti05] Douglas R. Stinson. *Cryptography : Theory and Practice, (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC Press, New York, November 2005.

[Swe96] W. Sweldens. The lifting scheme : A costum-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2) :186–200, 1996.

[SZL95] Guobin Shen, Bing Zeng, and M.-L. Liou. Adaptive vector quantization with codebook updating based on locality and history. *IEEE Transactions on Image Processing*, 12(3) :2003, 283-295.

[SZT04] R. Safabakhsh, S. Zaboli, and A Tabibiazar. Digital watermarking on still images using wavelet transform. In *ITCC 2004*, volume 1, pages 671–675. IEEE International Conference on Information Technology, 2004.

[Tan96] L. Tang. Methods for Encrypting and Decrypting MPEG Video Data Efficiently. In *ACM Multimedia*, pages 219–229, 1996.

[TC04a] S. P. Trivedi and R. Chandramouli. Locally most-powerful detector for secret key estimation in spread spectrum image steganography. In *Security, Steganography, and Watermarking of Multimedia Contents*, pages 1–12, 2004.

[TC04b] H. W. Tseng and C.C. Chang. High Capacity Data Hiding in JPEG-Compressed Images. *Informatica, Lith. Acad. Sci.*, 15(1) :127–142, 2004.

[TL03] C.-C. Thien and J.-C. Lin. A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function. *Pattern Recognition*, 36(12) :2875–2881, 2003.

[TM01] David S. Taubman and Michael W. Marcellin. *JPEG 2000 : Image Compression Fundamentals, Standards and Practice.* Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[TPC05] J. L. Toutant, W. Puech, and Fiorio C. Amélioration de l'invisibilité par adaptation de la quantification aux données à insérer. In *GRETSI'05*, 2005.

[UIO02] T. Uto, M. Ikehara, and M. Okuda. Wavelet packet algorithm for quadtree-based embedded image coding. In *ICASSP - IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 4, pages 3517–3520, 2002.

[Upm02] D. Upman. Jpeg-Jsteg Computer Software. Technical report, Proton World Int.l, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, ftp ://ftp.funet.fi/pub/crypt/steganography, 2002.

[VKG95] R.A. Vander Kam and R.M. Gray. Lossy compression of clustered-dot halftones using sub-cell prediction. In *DCC - Proc. Data Compression Conference*, volume 1, pages 112–121, 1995.

[Wat93] A. B. Watson. DCT quantization matrices visually optimized for individual images. In *Proc. SPIE*, volume 1913, page 202–216, 1993.

[WAW00] Peter H. W. Wong, Oscar C. Au, and Justy W. C. Wong. Image Watermarking Using Spread Spectrum Technique in Log-2-Spatio Domain. In

ISCAS00, Geneva, Switzerland, May 2000. IEEE International Symposium on Circuits and Systems.

[WBL02] Z. Wang, A. C. Bovik, and L. Lu. Why is image quality assessment so dificult ? In *In IEEE Conf. on Acoustics, Speech and Signal Processing*, volume 4, page 3313–3316, May 2002.

[WBSH05] Kevin M. Whelan, Felix Balado, Guenole C. M. Silvestre, and Neil J. Hurley. Iterative Estimation of Amplitude Scaling on Distortion Compensated Dither Modulation. In *Procedings of SPIE*, volume 5681 of *Security, Steganography and Watermarking of Multimedia Contents VII*, San Jose, CA, USA, January 2005.

[Wil94] J. Wilkins. *Mercury : or the Secret and Swift Messanger : Shewing, How a Man May With Privivacy and Speed Communicate His Thoughts to a Frind at Any Distance.* Rich Baldwin, London, 1694.

[WLW03] B. Wei, D. Liu, and X. Wang. Activity attack on Rijndael. In *Int. Conf. on Advanced Information Networking and Applications*, pages 803–806, 2003.

[WM01] P. W. Wong and N Memon. Secret and public key image watermarking schemes for image authentication and ownership verification. *IEEE Transactions on Image Processing*, 10(10) :1593–1601, Octobre 2001.

[WSB05] C. M. Wee, P.R. Sutton, and N.W. Bergmann. An FPGA network architecture for accelerating 3DES - CBC. In *Int. Conf. on Field Prog. Logic and Applications*, volume 1, pages 654–657, 2005.

[WSS00] M. J. Weinberg, G. Seroussi, and G. Sapiro. The LOCO-I Lossless Image Compression Algorithm : Principles and Standardization into JPEG-LS. *IEEE Trans. Image Processing*, 9 :1309–1324, 2000.

[WSZ+02] J. Wen, M. Severa, W. Zeng, M. Luttrell, and W. Jin. A Format-Compliant Configurable Encryption Framework for Access Control of Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6) :545–557, 2002.

[WYSV97] A. B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor. Visibility of wavelet quantization noise. *IEEE Trans. Image Proc*, 6(8) :1164–1175, 1997.

[WZY98] Zhu W., Xiong Z., and Zhang Y. Multiresolution watermarking for images and video : a Unified approach. In *IEEE-ICIP'98*, volume 1, pages 465–469, 1998.

[XCG97] Xia X., Boncelet C., and Arce G. A Multiresolution Watermark for Digital Images. In *IEEE-ICIP'97*, volume 1, pages 548–551, Santa Barbara, USA, 1997.

[YG05] Z. Yu and Y. Guan. A key pre-distribution scheme using deployment knowledge for wireless sensor networks. In *IPSN 2005 -Information Processing in Sensor Networks*, pages 261–268, 2005.

[YLLS00] G. Yu, C. Lu, H. Liao, and J. Sheu. Mean Quantization Blind Watermarking for Image Authentication. In *Proc. IEEE Int. Conf. on Image Processing*, volume III, pages 706–709, 2000.

[ZL77] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23 :337–343, 1977.

[ZL99] W. Zeng and S. Lei. Efficient Frequency Domain Video Scrambling for Content Access Control. In *ACM Multimedia, Orlando, FL, USA*, pages 285–293, Nov. 1999.

[ZSW04] Z. Zhang, Qibin Sun, and W.-C. Wong. A novel lossy-to-lossless watermarking scheme for JPEG2000 images. In *ICIP'04*, volume 1, pages 573–576. International Conference on Image Processing, 2004.

Table des figures

1.1 Principe de chiffrement asymétrique 20
1.2 Cryptage hybride. 24
1.3 Cryptage et décryptage synchrone 26
1.4 Cryptage et décryptage asynchrone. 27
1.5 Cryptage et décryptage du réseaux de Feistel 28
1.6 L'algorithme AES 29

2.1 Techniques de dissimulation d'information. 36
2.2 Problème classique de communication secrète. 37
2.3 Évaluation du PSNR comme mesure de qualité visuelle. (a) Image originale (b) Image fortement contrastée PSNR=24,63 dB, (c) Image fortement comprimée PSNR=28,80 dB. 40

3.1 Algorithme d'Huffman. 56
3.2 Décomposition en quadtree. 60
3.3 Décomposition en plans binaires des bits de poids forts jusqu'aux bits de poids faibles. 61
3.4 Compression par fractales. 62
3.5 Compression JPEG. 63
3.6 Bloc DCT et parcours zigzag. 64

4.1 Plan général de la méthode. 74
4.2 a) Image originale, b) ISP(4-1), c) ISP(8-5). 75
4.3 a) Partie de l'image originale, b) Même partie dans ISP(4-1), c) Même partie dans ISP(8-5). 76
4.4 La représentation du bloc B_8. 77
4.5 La représentation du B_{16}. 77
4.6 Les trois parcours dans ISP(8-5) a) Par ligne b) Par colonne c) spirale. . 77
4.7 Schéma d'insertion de données cachées et découpage en blocs de 4 bits. . 79
4.8 Brouillage / cryptage sélectif. 80
4.9 a) Image originale KHVomique (446 x 415), b) Image crypto-tatouée avec une capacité maximale W, c) Image crypto-tatouée avec une compression maximale, d) Histogramme de l'image originale, e) Histogramme de l'image (b), f) Histogramme de l'image (c). 83

4.10 a) Image originale colposcopie (446 x 415), b) Image crypto-tatouée avec une capacité maximale W, c) Image crypto-tatouée avec une compression maximale, d) Histogramme de l'image originale, e) Histogramme de l'image (b), f) Histogramme de l'image (c). 84
4.11 a) Image médicale échographique, avec de grandes zones homogènes, b) Image médicale cryptée par bloc avec AES mode ECB. 87
4.12 a) Image originale de Lena, b) Image cryptée avec AES par bloc de 128 bits, c) Permutation de régions de l'image cryptée, d) Copie d'une région de l'image cryptée et collage sur une autre zone, e) Décryptage de (c), f) Décryptage de (d). 88
4.13 Combinaison d'un cryptage à clef secrète, d'un cryptage à clef publique et d'une méthode d'IDC. 90
4.14 Cryptage et décryptage asynchrone. 91
4.15 a) Image cryptée avec l'algorithme de chiffrement par flot et bruitée (TEB=$1,95\ 10^{-3}$), b) Image cryptée avec l'algorithme de chiffrement par flot et bruitée (TEB=$1,25\ 10^{-1}$), c) Résultat du décryptage de l'image figure (a), d) Résultat du décryptage de l'image figure (b). 94
4.16 a) Image originale, b) Image cryptée par flot avec une clef de 128 bits, c) Image (b) marquée avec la clef secrète cryptée, d) Différence entre les images (b) et (c), e) Décryptage de l'image (c), f) Différence entre l'image originale (a) et (e). 96
4.17 a) Image Lena chiffrée avec l'algorithme AES en mode ECB et marquée, b) Image Lena chiffrée avec l'algorithme AES en mode CFB et marquée, c) Résultat du décryptage de (a), d) Résultat du décryptage de (b), e) Différence entre l'image originale et (c), f) Différence entre l'image originale et (d). 98

5.1 L'algorithme JPEG. 105
5.2 Le mode CFB de l'algorithme AES. 108
5.3 Présentation générale de la méthode proposée. 110
5.4 a) Image originale, b) Cryptage des coefficients DC, c) Remplacement des DCs par zéro. 112
5.5 a) Bloc homogène, b) Coefficients DCT quantifié de (a), c) Bloc texturé, d) Coefficients DCT quantifié de (c). 113
5.6 a) Image médicale originale d'un cancer du colon, b) Image cryptée pour $C = 128$, c) Image cryptée pour $C = 8$, d) Décryptage d'une région à 100%. 117
5.7 a) Image médicale d'un scanner 512×512 pixels, b) Image cryptée pour $C = 128$, c) Image cryptée pour $C = 8$, d) Décryptage de deux régions à 50%. 118
5.8 a) Peinture numérique originale 512×640 pixels, b) Image cryptée pour $C = 128$, c) Image cryptée pour $C = 8$, d) Décryptage de deux ROIs. . . 120
5.9 a) Peinture numérique originale 640×480 pixels, b) Image cryptée pour $C = 128$, c) Image cryptée pour $C = 8$, d) Décryptage d'une région. . . . 121

5.10 a) Image originale #083 de la séquence, b) Image (a) cryptée pour $C = 128$, c) Image originale #135 de la séquence, d) Image (c) cryptée pour $C = 128$. 123
5.11 a) Région de l'image figure 5.10.c, b) Région de l'image figure 5.10.d. . . 123